Object Localization from RGB-D Images and Spatial Referring Expressions

by

Cecilia Mauceri

B.A., Oberlin College, 2013

M.S., University Illinois Urbana Champaign, 2015

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2021

Committee Members:

Christoffer Heckman, Chair

Martha Palmer

Jim Martin

Alessandro Roncone

Nisar Ahmed

Mauceri, Cecilia (Ph.D., Computer Science)

Object Localization from RGB-D Images and Spatial Referring Expressions

Thesis directed by Prof. Christoffer Heckman

Multimodal learning uses inputs from multiple modalities, for example text and vision, to support a machine learning objective, such as automatically locating an object based on its description. Learning from multiple sensors or multiple information representations can produce better results than a single modality as they reinforce and complement each other. Additionally, some tasks are inherently multimodal. For example, humans perceive the world in a different modality than they describe it. Therefore any communication-based tasks require multimodal representations and learning.

The research challenge for multimodal learning is setting up a learning system with diverse, heterogeneous data from different modalities, so that correlations can be effectively learned. This thesis explores two applications of multimodal learning; RGB-D semantic segmentation and referring expression understanding. In the former, depth measurements of the scene are used together with color photos to label pixels belonging to different object classes. In the latter, multimodal representations of images and text support the generation and understanding of object descriptions.

Acknowledgements

Thank you, Chris, for being an outstanding advisor. In my mind you won the award, even if I couldn't convince the Graduate School to confer it.

Thank you, Martha, for providing your insightful perspective from the other side of the cross-modal divide.

Thank you to my committee, for all your feedback along this journey.

Thank you, John, Sina, Fernando, and Mike Kasper, for being excellent role models and friends as the senior PhDs when I started.

Thank you, Mary and Xuefei, for your continued interest in my research and for taking up the multimodal torch.

Thank you, ARPG, for keeping the conversation going and fresh every semester with new projects and papers to discuss.

Thank you, Mom and Dad, for encouraging me to start doing computer science research in undergrad. I know you told me that the PhD was optional, but I also wouldn't be here without your incitement and support.

Thank you, Steffen. Your support was the most important of all. We bounced ideas off each other at breakfast, lunch, and dinner, but you also kept my work-life balance in check by encouraging me to take evenings and weekends off. You are my best proofreader, and your graphics skills are also invaluable. Thank you for going on the PhD adventure with me.

Contents

Chapter

1 Introduction **1**

2 An Introduction to RGB-D Fusion **5**

2.1 Depth Sensing 7

2.1.1 Stereo Matching 7

2.1.2 Structured Light Sensors 8

2.1.3 Time of Flight Sensors 9

2.2 Monocular Depth Prediction 11

2.3 RGB-D Segmentation Networks 12

3 Multi-modal sensing in the dark **14**

3.1 Related Work 16

3.2 Method 17

3.2.1 Datasets 17

3.2.2 Synthetically Darkening Images 18

3.2.3 Semantic Segmentation Networks 21

3.3 Results 23

3.3.1 RGB-D Backbone Selection 23

3.3.2 Feature Exploration 25

3.3.3 Low-light Results 27

3.4 Conclusion 29

4 RGB-D Artifact Detection for the Subterranean Challenge **33**

4.1 SubT Artifact Detection 33

4.2 Method 37

4.2.1 Model 37

4.2.2 Data 38

4.3 Conclusion 41

5 An Introduction to Referring Expressions **43**

5.1 Early Referring Expressions Work 44

5.2 Referring Expressions Datasets 46

5.2.1 Synthetic datasets 47

5.2.2 Realistic Datasets 49

5.3 Data-driven Referring Expression Generation 51

5.3.1 Generation Evaluation 55

5.4 Data-driven Referring Expression Comprehension 56

5.4.1 Ranking Image Regions with Generative Models 57

5.4.2 Direct prediction 60

5.4.3 Comprehension Evaluation 65

5.5 Conclusion 66

6 A New Dataset for Spatial Referring Expressions **68**

6.1 SUN-SPOT 70

6.1.1 Expression Collection 72

6.1.2 Landmark Object Annotation 72

6.1.3 Dataset Statistics 75

6.2 Experiments 76

6.2.1 Models 76
6.2.2 Referring Expression Generation 79
6.2.3 Referring Expression Comprehension 81
6.2.4 Effects of Depth 81
6.3 Conclusion 82

7 Conclusion **83**

Bibliography **85**

Appendix

A Glossary **96**
A.1 Glossary 96

B Code Repositories **98**
B.1 Data Management Tools 98
B.1.1 SUNRGBD to COCO annotation format 98
B.1.2 RGB-D Data Loaders 99
B.1.3 SUN-Spot Data Loader 99
B.2 Data Annotation Tools 99
B.2.1 SUN-Spot Image/Caption Server 99
B.3 Data Processing 99
B.3.1 Lidar to Image Plane Projection 99
B.3.2 Monocular Depth Prediction 100
B.3.3 Lidar Completion 100
B.4 Networks 100
B.4.1 Depth Aware CNN 100

B.4.2 RGB-D Semantic Segmentation 100

B.4.3 RGB-D Object Detection 101

B.4.4 Referring Expression Generation and Comprehension 101

B.4.5 CNN Visualization 101

Tables

Table

3.1 Quantitative results on Scenenet without artificial darkening 23

3.2 Quantitative results on Cityscapes without artificial darkening 25

4.1 Evaluation of depth completion error using IP-Basic with different smoothing methods. RMSE is Root Mean Squared Error. MAE is Mean Absolute Error. The error metrics treat the stereo-depth (SubT-Tunnel) or depth measured using a structured light depth sensor (SUNRGBD) as the ground truth. The unit used is meters. 40

5.1 TUNA's attribute sets [1] 47

5.2 A comparison of three popular referring expressions datasets using the MS-COCO [2] images. 51

5.3 The results of all generation models evaluated on ReferIt[3] and Google RefExp[4] using the automated similarity metrics ROUGE[5] or METEOR[6]. 55

5.4 The results of all comprehension models that evaluated on ReferIt [3] or Google RefExp [4] datasets. 65

6.1 A comparison of SUN-SPOT to three popular referring expressions datasets. While SUN-SPOT is an order of magnitude smaller than these other datasets both in the number of images (Images), and the number of referring expressions (REs), it has a higher frequency of location prepositions (LocPrep), longer average expressions (Length), and more object class labels (Classes). 75

6.2 Quantitative results for generation and comprehension on RGB models. Columns are BLEU1 (B1), ROUGE-L(R-L), CIDEr(C), and Precision at 1(P@1). The first two rows compare ground truth referring expressions to establish an upper-bound. The other rows evaluate generated sentences. 80

6.3 Results comparing RGB image features (VGG and VGG+fine) to RGB-D image features (VGG+D and VGG+D+fine). Metrics are the same as used in Table 6.2. . 82

Figures

Figure

2.1 Depth can help disambiguate concave and convex surfaces. In the illusion 2.1a, the shape can be interpreted as a pyramid on top of a cuboid or hole through the center of the cuboid. In the depth map representation, there is a clear interpretation of the 3D geometry. The pyramid has no depth discontinuities at the front edge, but the hole has a sharp depth discontinuity at the front edge. 6

2.2 For mass produced products, standardized containers can be a cost saving measure, but the consumer must rely on visual cues to discriminate between the products. This can be an issue for blind users who rely on 3D geometry to guide their object recognition. A common coping mechanism is to change the 3D geometry of one product by adding for example a rubber band as seen on the conditioner bottle on the right. Similarly, an object classification system which relied on depth alone would be unable to distinguish between these product pairs. 6

2.3 The KITTI dataset provides lidar depth measurements, "ground truth" semi-dense depth using a combination of lidar and SGM [7], and stereo-pairs which can be used to calculate SGM directly. This figure shows a comparison of the three methods of measuring depth. The same legend applies to all three depth maps. 10

3.1 Comparing an RGB image with good exposure (first row) and an RGB image darkened with a physic-based model to imitate low illumination (second row) reveals the data challenges for semantic segmentation in dark environments. The detail, in the second column, shows an enlarged crop of the red bounding box on the example image. For the darkened image, we have brightened the detail to reveal the increased noise. In addition to noise, pixel intensities in the darkened image are skewed towards zero. The median pixel value changes from 72, in the original, to 7 in the darkened image. The increased noise and decreased dynamic range make RGB semantic segmentation networks more prone to error. The depth map, shown in the third row, has fewer semantic features, but is invariant to illumination, providing a stable modality in dark environments. 15

3.2 Grid shows a range of gain and gamma correction applied to an example image from Cityscapes. Top-right is the original image. The ranges represented, $G = [0.5, 1.0]$, $\gamma = [1.0, 2.2]$ are the same ranges used in training augmentation. 19

3.3 Grid shows re-brightened images from Figure 3.2 to illustrate the noise introduced by artificial darkening. Each image has an enlarged insert in the bottom right corner to show a close up of noise. 19

3.4 Histogram of RGB pixel magnitude under different darkening treatments. Percentage is estimated from a sample of 10,000 pixels. Clipping of the distribution toward zero pixel magnitude indicates underexposure leading to a loss of image information. The legend on the right is used for both plots. 20

3.5 Examples of segmentation using different architectures. The first column is the input. The second column is the ground truth segmentation. The following columns are labeled with the network name. White pixels have no ground truth label and are masked during evaluation. The DepthAware network shows a failure mode in the third row. 24

3.6 Activations after the backbone CNN for three object classes visualized using Grad-CAM [8]. The RGB and RGB-D models are learning different feature distributions. For example, the road activations show that the RGB-D model's feature applies to the whole road surface, while the RGB model 26

3.7 Bar graphs show the mIoU performance on the Cityscapes dataset with gamma-correction (γ) and gain (G) applied at test time. One example image is shown for each condition. All conditions have added Gaussian noise ($\sigma = 0.01$) and Poisson noise. 28

3.8 Bar graphs show the mIoU performance on the SceneNet dataset with gamma-correction (γ) and gain (G) applied at test time. One example image is shown for each condition. All conditions have added Gaussian noise ($\sigma = 0.01$) and Poisson noise. 28

3.9 In an input ablation study, we compare network mIoU performance on full RGB-D input to input containing only a subset of channels. For each condition, the ablated channels are set to zero. The same legend is used for both datasets. 30

3.10 We run a sensitivity analysis where one darkening variable is perturbed at a time. Networks trained with low-light augmentation are invariant to Gaussian noise. For all other variables, darker values decrease network performance. The same legend is used for both datasets. 31

4.1 This figure shows all final event artifacts which can be localized using a visual sensor. Artifacts 4.1f-4.1i have additional detectable signals including heat(4.1f and 4.1g), Bluetooth(4.1h and 4.1i), and WiFi(4.1h). Not pictured is the CO_2 gas artifact which cannot be detected visually [9]. 34

4.2 Artifact detection is scored on thresholded euclidean distance. The reported position of the artifact localization point must lie within 5 meters of the ground truth position to be considered a valid report [10]. 34

4.3 An example of the difference between lidar and stereo-depth from the SubT-Tunnel dataset. Color indicates depth in meters. Even for pixels where overlapping measurements are available, the two methods of measuring depth have a 1.75 meter mean absolute difference. Lidar and difference pixels are enlarged for visualization. 38

4.4 An example of SUNRGBD data. The depth channel is captured with a structured light sensor, in this case, an Xtion Pro. The IP-Basic depth completion is based on a 5% random sample of the depth channel. Error in IP-Basic is most apparent at the edges of objects. 39

4.5 These three plots show confusion matrices for various depth completion techniques. Each (x, y) pixel represents the frequency that depth x was predicted to be depth y. Figure 4.5a shows the confusion between stereo and lidar depth measurements. We see that stereo measurements compressed range of depth compared to lidar. We expect to see that Figures 4.5b and 4.5c follow a similar distribution, but denser. Instead, we see that Figure 4.5b has many artifacts. 41

5.1 Synthetic referring expressions dataset collection interfaces. In 5.1a, the annotator is asked to describe the objects in the red boxes. In 5.1b, the annotator is asked to describe the object pointed to by the black arrow, and in 5.1c, the annotator is asked to describe the object pointed to by the blue arrow. 46

5.2 Examples from CLEVR-Ref+ dataset [11] 48

5.3 Referring expressions and scene examples from realistic datasets 50

5.4 Datasets which provide referring expressions in sentence context 52

5.5 Mao et al.'s baseline referring generation architecture [4] 52

5.6 The differences between object segmentation, instance segmentation, and referring expression comprehension. Object and instance segmentation methods are only capable of distinguishing between a predefined vocabulary of object classes, while referring expressions include a much wider and more granular range of object descriptions. Figure from [12] 56

5.7 Network diagrams illustrating the differences and similarities between [12] and [13]. Both use an image CNN to encode the input image and a LSTM to encode the text features. Both use final layers which include a convolutional classification layer to label each pixel followed by deconvolutional upsampling. Figure from [13] 62

5.8 Results from Hu et al.'s CMN. Ground-truth is shown in the first column, followed by the predicted regions. The referent is in the red solid box and landmark is in the green dashed box. The third column visualizes attention weights where darker is higher attention. Figure from [14] 63

6.1 An example image from the SUN-SPOT dataset including three objects with labeled referring expressions. We annotate objects which have multiple class instances in an image (i.e. the flowers in the green and yellow bounding boxes) to test the ability of referring expression systems to differentiate between objects. We also annotate classes which occur rarely (i.e. the calendar in the blue bounding box) to balance the frequency of object classes and test zero-shot learning. We collect multiple referring expressions for each annotated object. A selection of annotations for the three highlighted objects in this image are shown on the right. 71

6.2 SUN RGB-D annotations for the image shown in Figure 6.1. From left to right, we show the depth map (blue is closer to the camera, yellow is farther), a 3D projection of the RGB-D image with overlaid 3D object bounding boxes, instance segmentation (each color labels an object instance), and semantic segmentation (each color labels an object class). 71

6.4 Word cloud of the most frequent vocabulary in our dataset. The vocabulary focuses on objects and locations. 76

6.5 Relative frequency of the 10 most frequent location prepositions in Google RefExp and SUN-Spot 77

Chapter 1

Introduction

In the past, Computer Vision and Natural Language Processing have been approached as separate areas of research. This made sense because different data benefits from different learning approaches; for example, sequential data like text needs learning machinery that can learn sequence-based dependencies whereas object detectors in images benefit from being spatially invariant and able to detect objects independently of their position in the image. However, while computer vision and natural language processing have made great strides on their chosen tasks, data doesn't always conform nicely to the silos these research areas define. We often want to be able to talk about images using language, including giving instructions relative to the environment, captioning images for the visually impaired, or engaging in dialog about things we see. Additionally, even within one of these domains, the data may come from different sources with very different characteristics, for example, video data does have time dependencies, but individual photos do not.

Multimodal learning is an umbrella term for machine learning tasks which bridge data representations. In this context, modality refers to the type of information and its representation. At the intersection of Computer Vision and Natural Language Processing, multimodal learning includes tasks like image captioning, visual question answering, visual language navigation, and referring expression understanding. These are tasks where both a detailed semantic understanding of the visual environment and a detailed semantic understanding of language is required. In recent years, interest in multimodal learning has exploded with top conferences including topics of interest such as "Speech, Vision, Robotics, Multimodal Grounding" at the Conference on Empirical Methods in

Natural Language Processing and "Vision + language, vision + other modalities" at the Conference on Computer Vision and Pattern Recognition. This attention is rightfully earned by the field as explores new important applications of machine learning.

There are three main motivations for pursuing multimodal learning. First, modalities can complement each other, with one modality filling information gaps in the other. For example, hearing can alert us to movement outside our field of vision or in a visually occluded area. While with vision, we localize objects more precisely than using hearing. This complementary nature is why emergency vehicles use both sirens and lights. The second motivation for multimodal learning is that modalities can reinforce each other, increasing the model's confidence in its prediction. A good everyday example for reinforcement is baking bread. Typically, as a loaf approaches done-ness, a heavenly smell begins to spread, the color darkens, and the bread begins to sound hollow when gently rapped with a fist. All three indicators are weak predictors of done-ness, but together, they help the baker decide when to remove the bread from the oven. Finally, multimodal learning introduces abstractions that allow more sophisticated communication and reasoning about the data. For example, humans perceive the world using their senses, but they describe it using language and reason about it symbolically. Therefore any communication-based tasks require translating between sensory input, semantic representations, and language.

This thesis concentrates on two separate applications of multimodal learning. The first is an application where the modalities complement and reinforce each other. In this project, I use multiple visual sensors to increase the accuracy of semantic segmentation and object detection in dark environments. The second project is an application where communication about an environment is the goal. In this project, the system learns to understand and generate visual descriptions of objects. Serendipitously, in the second application, multiple visual sensors also prove helpful for scene understanding, so the first project also benefits the second.

Perception systems relying on commercial cameras often suffer in dark environments. In Chapters 2, 3, and 4, I discuss how using RGB-D image input in low-illumination conditions can increase the robustness of semantic segmentation and object detection. Chapter 2 introduces various

depth sensors and prior work in RGB-D semantic segmentation. In Chapter 3, I test several semantic segmentation architectures with different depth fusion methods over a range of lighting conditions. To provide sufficient training and test data, I artificially darken the scenes using a physics-based model. For dark scenes, semantic segmentation with RGB-D performs approximately twice as well as RGB without the additional depth channel. These results show the clear advantage of utilizing an additional depth sensor on platforms operating in low-light environments. In Chapter 4, I extend these results to the system used in the DARPA Subterranean Challenge, which uses an on-board lidar instead of a depth sensor.

The RGB-D image representations also prove useful for understanding spatial referring expressions. In Chapter 5, I introduce the task of referring expression understanding, and in Chapter 6, I introduce a new dataset for localizing objects using spatial referring expressions. The purpose of this dataset is to aid human robot collaboration in scenes which provide challenges to object detection and localization due to clutter, occlusion, unknown object classes, and multiple instances of the same object class. It also presents a challenging dataset for natural language and the understanding of spatial prepositional phrases in English. The dataset provides 7,987 referring expressions of 1948 images with an average of 2.6 spatial prepositions per expression. I show that existing referring expression methods do not transfer well to this domain, and provide baseline performance for this localization task.

Both of these projects face the challenge of learning from heterogeneous data. In the RGB-D fusion networks, the challenge is that depth measurements are smoother with less high-frequency information than color images. Additionally, lidar produces a sparser signal than cameras. The research challenge is how to integrate these depth measurements with conventional color semantic segmentation and object detection networks. For referring expressions understanding, the challenge is how to evaluate if networks are learning to understand the correspondence between the referring expression and the scene semantics. If there are dataset biases that make it easy to guess the target object, the network may give the impression of understanding complex spatial relationships when it is in fact representing much simpler correspondences between the text and image, such as object

class detection. In this work, I show that both the system design and evaluation are crucial to successful multimodal learning.

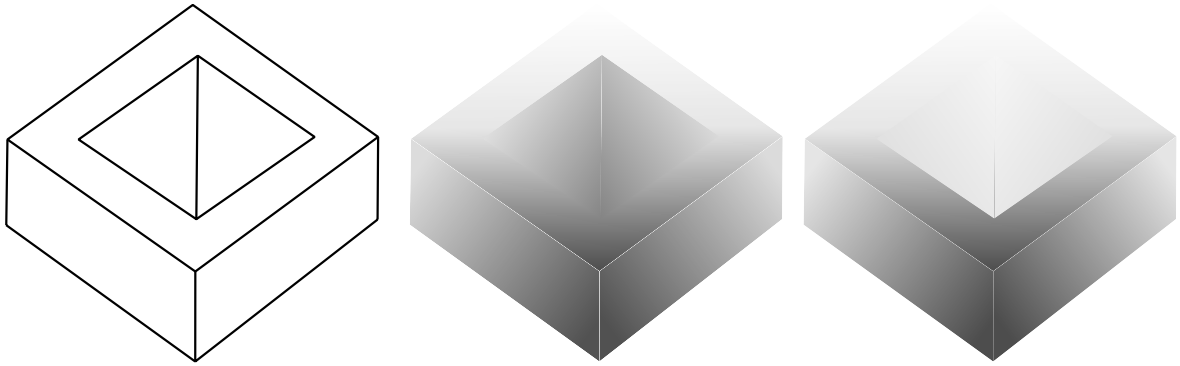
Chapter 2

An Introduction to RGB-D Fusion

Depth contains clues about occlusion, object shape and pose, and can help distinguish a concave surface from a convex one. Figure 2.1 shows a classic optical illusion where depth would help disambiguate between two geometric interpretations. Based on similar observations, many have hypothesized that using depth, in addition to color, should improve performance on many visual tasks. Consequently, depth has been used for segmentation, pose detection, and navigation, among other applications. These three example tasks rely on understanding 3D geometry. For segmentation, the edges of objects have 3D discontinuities that aid edge detection. Finding the 3D orientation of the object is the goal of pose detection. Therefore, the distance to obstacles is an important consideration. Depth provides a direct measurement of the 3D geometry of the scene.

Visual light, on the other hand, captures details of surface texture, color, and other semantic information that is often not resolved or available in the depth modality. Figure 2.2 shows examples of objects that are identical from their 3D geometry but have important visual differences. A system which relied on depth alone would be unable to discriminate between these objects. In addition to having rich semantic features, visual features often correlate with geometric features. For example, a geometric edge is often also present as a color edge. Therefore, for tasks such as object detection and classification, state-of-the-art networks can use 2D information and color alone with great success.

For segmentation problems, the geometric information provided by depth and surface texture provided by visual sensors like commercial cameras are complementary signals. Depth discontinuities often correspond to object boundaries and can help refine crisp segment edges. Color features help



(a) Visual Illusion (b) Depth Map of Pyramid (c) Depth Map of Hole

Figure 2.1: Depth can help disambiguate concave and convex surfaces. In the illusion 2.1a, the shape can be interpreted as a pyramid on top of a cuboid or hole through the center of the cuboid. In the depth map representation, there is a clear interpretation of the 3D geometry. The pyramid has no depth discontinuities at the front edge, but the hole has a sharp depth discontinuity at the front edge.

Figure 2.2: For mass produced products, standardized containers can be a cost saving measure, but the consumer must rely on visual cues to discriminate between the products. This can be an issue for blind users who rely on 3D geometry to guide their object recognition. A common coping mechanism is to change the 3D geometry of one product by adding for example a rubber band as seen on the conditioner bottle on the right. Similarly, an object classification system which relied on depth alone would be unable to distinguish between these product pairs.

identify the object class. The challenge is learning joint representations from these two signals which allow downstream segmentation tasks to benefit from the different sensors.

This chapter provides a literature review of multi-modal depth and visual learning focusing on RGB-D segmentation. In Section 2.1, I introduce depth-sensing methods. In Section 2.2, I discuss networks which translate between RGB and depth modalities such as monocular depth prediction and pseudo-lidar networks, and finally, in Section 2.3, I discuss joint depth-visual representations as they have been used in RGB-D segmentation.

2.1 Depth Sensing

There are three main methods of measuring depth, stereo matching, structured light sensors, and time-of-flight sensors.

2.1.1 Stereo Matching

Humans have depth vision because we have two eyes. Each eye observes a slightly offset version of the same scene. In the two offset observations of the scene, objects at different distances from the viewer shift relative to each other. This shift, or *parallax*, allows the brain to triangulate the distance to the objects.

Stereo matching operates on the same principle. The eyes of the human observer are replaced with two cameras, or a stereo-pair. The distance, or baseline, between the two cameras is fixed, and usually for simplicity of calculation, the cameras are oriented in such a way that images captured are parallel and are horizontally aligned, or *rectified*. By matching pixels in each image, the distance, Z, to the object can be calculated using the disparity, d, between the pixels in the left and right images, the focal length in pixels, f, and the baseline, B.

$$Z = fB/d \tag{2.1}$$

The focal length and baseline are directly observable from the camera configuration. The remaining challenge is estimating the disparity map, or the pixel correspondences between the two

images. Many algorithms exist for estimating the disparity map. One which is widely used, and will crop up again in Chapter 3 of this dissertation is semi-global matching (SGM) [15]. SGM is used to generate disparity maps for the Cityscapes dataset [16] and for the depth completion task on the KITTI dataset [7].

Because a depth measurement is calculated for each pixel pair, the result of stereo matching is usually a dense depth map, or a matrix with the same dimensions as one of the input images where each pixel has been replaced with a depth measurement. The image-aligned nature of the depth map makes it convenient as a starting point for joint visual-depth representations.

Stereo matching algorithms are prone to a variety of errors. They rely on surface features and produce noisy maps in feature-poor environments. They also struggle with motion blur and directional lighting changes such as specularities. Additionally, depth can only be triangulated for points observable by both cameras. Scene occlusion may lead to gaps in the depth map. These often appear like shadows at object edges. Finally, the precision at which depth can be observed is limited by the baseline of the camera pair. As objects move farther and farther away, the observation angles approach parallel lines and the disparity decreases to zero.

2.1.2 Structured Light Sensors

Structured light sensors project a known pattern on the scene and measure the distortion of that pattern to determine the 3D geometry of the scene. The actual depth calculation is similar to stereo matching with the pattern projector standing in for one of the cameras and the light sensor standing in for the other camera in the camera pair. The advantage of structured light is that even on feature-less or feature-poor surfaces, the distortion of the light pattern provides a reliable disparity measurement.

While the light pattern addresses some of the challenges of feature poor surfaces, it is not a panacea. Depending on the wavelength used, environmental lighting can obscure the projected pattern. For example, sunlight can obscure an active infrared pattern. In addition, multiple projectors in the same space will produce pattern interference which degrades the depth measurements [17].

In Chapter 6, I use imagery from the SUN-RGBD[18] dataset for referring expression understanding. SUN-RGBD provides depth maps collected using a variety of structured light sensors including the Kinect v1, Intel RealSense, and Asus Xtion LIVE PRO. These sensors are convenient for indoor applications where the effective depth range required is low. They provide an out-of-the-box depth solution for low cost.

2.1.3 Time of Flight Sensors

Time-of-flight sensors measure the time that it takes a light signal emitted from the sensor to bounce off an object in the scene and return to the sensor. Lidar is a time-of-flight sensor using lasers which has been particularly successful in autonomous driving applications. Lidar, and time-of-flight sensors in general, provide more precise depth measurements than stereo or structured light methods. However, they have a larger form factor, lower resolution, and are more expensive. Lidar comes in 2D and 3D configurations. The output of a 3D lidar is a 3D point cloud which can be projected onto the camera plane to produce a sparse depth map.

Lidar point clouds have high accuracy for measurement points but can accumulate some error from the scanning process. The most commonly used lidar for automated driving have mechanical spinning components to steer the laser to different parts of the field of view. A Velodyne lidar, for example, has a vertical stack of lasers which spin at 15Hz to cover a 360° field of view [19]. At high speeds, a single scan is distorted by the distance traveled from the beginning to the end of the rotation. For example, at 65 mph, a car will have traveled 6 ft during a single scan. Scan distortion errors can be eliminated by using a flash lidar, a technology which illuminates the whole scene simultaneously, but flash lidar have a much smaller effective range than mechanical lidar ($< 100m$ vs. $> 100m$), which is also a challenge for high speed operation [20]. Post-processing can also be used to un-distort the scan.

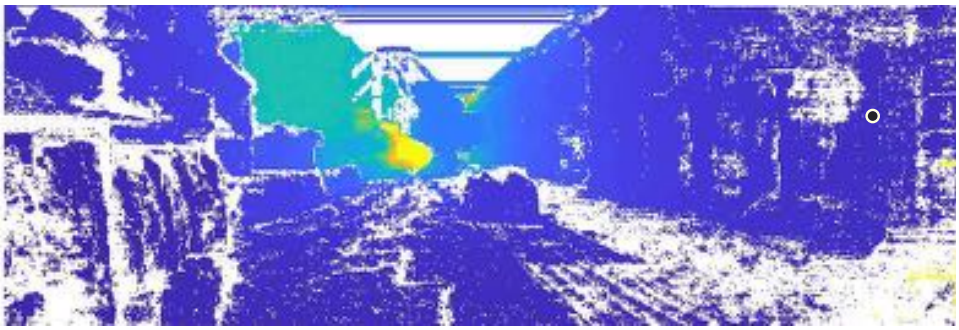
In addition to scan distortion, measurement error can occur on reflective or transparent surfaces which cause the laser return signal to be lost.

The biggest challenge in using lidar together with cameras is aligning the data. If the

(a) RGB

(b) Lidar

(c) Semi-Dense Ground Truth [7]



(d) SGM

Figure 2.3: The KITTI dataset provides lidar depth measurements, "ground truth" semi-dense depth using a combination of lidar and SGM [7], and stereo-pairs which can be used to calculate SGM directly. This figure shows a comparison of the three methods of measuring depth. The same legend applies to all three depth maps.

transformation between the lidar and camera is known, as well as the camera intrinsics, the 3D point cloud can be projected onto the camera plane. However, the result will be a sparse depth map. Conventional CNNs require dense input matrices, so additional post-processing is needed to use lidar-based depth maps with off-the-shelf object detectors.

The KITTI dataset introduced a depth completion benchmark for estimating dense depth from sparse lidar depth projections [7]. Figure 2.3 shows a comparison of lidar, ground truth depth, and SGM stereo depth from the KITTI dataset. The semi-dense ground truth is constructed by collating multiple lidar scans for increased density and using SGM depth measurements to remove large outliers due to reflective and transparent surfaces. It is interesting to observe how lidar and SGM can complement each other and be used to reduce error.

2.2 Monocular Depth Prediction

In 2014, Eigen et al. [21] published a method showing that it was possible to train a network to predict depth measurements from a single image. Subsequently, deep monocular depth estimation or depth prediction became a new standard task on benchmarks such as Cityscapes [16], KITTI [7], and Make3D[22] with many many competing approaches. Khan et al. [23] and Zhao et al. [24] provide good surveys of state-of-the-art methods.

While absolute error on predicted depth maps is very low, they have only seen limited application. It remains unclear why they have not proved more useful. One possibility is that information in the predicted depth is redundant with the information in the input RGB image. The network can only predict depth based on the features in the color image. If an object boundary is ambiguous in the color image, the network can only draw on its prior experience during training to predict its value. In many cases these priors prove accurate, but if the monocular depth prediction network can learn such priors, so could downstream tasks. Therefore, using monocular depth prediction as an intermediate task is potentially unnecessary.

Another possibility is that the predicted depth representation is not being used effectively by downstream tasks. Recently, pseudo-lidar has proved a better representation for monocular depth

prediction than depth maps[25, 26]. Pseudo-lidar is in essence monocular depth prediction with the output represented as a 3D point cloud. Wang et al. and Vianney et al. have shown that using a 3D point cloud pseudo-lidar representation improves 3D object detection. The success of the pseudo-lidar representation highlights the importance of representation in multimodal learning.

2.3 RGB-D Segmentation Networks

In 2011, the release of the Kinect and other low-cost commercial depth sensors galvanized the research community to explore applications for RGB-D. It seemed intuitive that adding depth as a sensor modality could improve segmentation results since edges often correlate with depth discontinuities, and indeed, many papers showed promising results. Fooladgar and Kasaei provide a good survey of methods on interior datasets which includes many deep convolutional methods developed specifically for RGB-D input [27]. However, on benchmark tasks with available depth maps, such as the "Cityscapes Pixel-Level Semantic Labeling Task," RGB methods often outperform RGB-D methods [28].

Part of this limitation results from the scant amounts of available RGB-D training data. To circumvent data limitations, Gupta et al. [29] proposed transferring weights learned on RGB images to depth networks. Networks trained on RGB images have three input channels; therefore, a three-channel depth representation was needed in order to transfer the weights. Gupta et al. proposed three hand-engineered channels, called HHA, as a 3-channel depth encoding scheme.

HHA stands for **H**orizontal disparity, **H**eight above the ground plane, and **A**ngle of surface normal. These quantities are calculated from the depth channel. For example, height above the ground plane is calculated by fitting vertical and horizontal surfaces to identify a gravity direction. The 3D point cloud is rotated so that gravity is vertical, and the ground plane is the lowest pixel in the new orientation frame [30].

HHA encoding has been widely used, but has several limitations. If there are not enough vertical and horizontal surfaces visible to identify the ground, the HHA encoding will be noisy. Additionally, low-level HHA features do not resemble RGB features. HHA images tend to be smooth

except at object boundaries. RGB images have more surface texture. This mismatch between RGB and HHA results in many filters going unused when RGB network weights are transferred to HHA networks [31].

One alternative to HHA is DepthAware convolutions [32]. DepthAware convolutions are a modification to the standard convolutional kernel. A DepthAware convolution weighs each convolution by the magnitude of the depth map at that position. Wang et al. [32] train their DepthAware convolutional neural network (CNN) from scratch and do not use transferred weights. While they show a 11% improvement over HHA methods trained from scratch (61% vs 50% accuracy), Gupta et al.'s original HHA method using transfer learning outperforms Wang et al. by a further 12% (73%). This emphasizes the limitations of small datasets—in this case, 795 training examples.

Another way to introduce depth to the network is as an additional input channel. Concatenating the three color channels and the depth channel produces a four-channel input. This type of network is sometimes called an early-fusion approach because the depth and image information are fused before features are extracted. Due to lack of training data, few experiments have been performed with early fusion RGB-D segmentation networks. Gupta et al. [29] experimented with such a network, but found that it compared unfavorably with their HHA results. Sanchez-Riera et al. [33] show that an early-fusion CNN outperforms a late-fusion CNN as well as RGB and depth features alone on an object recognition task. Sanchez-Riera et al.'s object dataset contains 250,000 images, providing an example of the scale of training data necessary for strong performance.

One benchmark dataset which does provide a large number of RGB-D images is Cityscapes, an autonomous driving dataset [16]. Among the state-of-the-art networks trained on Cityscapes, the depth channel is used for hard-negative mining [34], HHA fusion [35], and an innovative ResNet-inspired design[36]. However, the current published RGB leader [37] performs 2 percentage points better than the RGB-D leader [34] (83.4 vs 85.4 mIoU) underscoring again the challenge of using depth effectively.

Chapter 3

Multi-modal sensing in the dark

Not all operating environments for autonomous agents are well illuminated. From nighttime driving, to exploring tunnels and natural caves, to navigating indoor spaces, bright lighting is not always available. Even if an onboard light source is available (e.g. headlights), it is typically targeted in a specific direction, whereas cameras may be placed around the platform for panoramic coverage, outside the effective area of the light source. In dark environments, consumer cameras produce noisy images (See Figure 3.1). However, many semantic segmentation methods rely on well-lit scenes where object texture and geometry are clearly visible. As a result, semantic segmentation suffers in poor lighting conditions [38].

Luckily, many robotic platforms carry depth sensors, such as active IR or lidar, to assist in mapping the environment. These active depth sensors perform well in dark environments because they provide their own light source at non-visible wavelengths. As described in Chapter 2.3, through a variety of approaches, RGB-D images have been shown to incrementally improve segmentation over RGB alone in bright environments. However, adding an additional sensor to a platform comes at a cost, sometimes quite literally; a single lidar sensor can cost thousands of dollars. Other costs include additional payload weight, which is a valuable resource for small mobile platforms, and increased computation due to larger input data, which can hinder real-time operation. With advances in RGB semantic segmentation, accuracy is so high in state-of-the-art models that one might ask if marginal gains from using an active depth sensor justify the added payload and additional processing. We

This work was supported by the DARPA AIDA program award number FA8750-18-2-0016, and the DARPA Subterranean Challenge award number HR0011-18-2-0043, team "MARBLE."

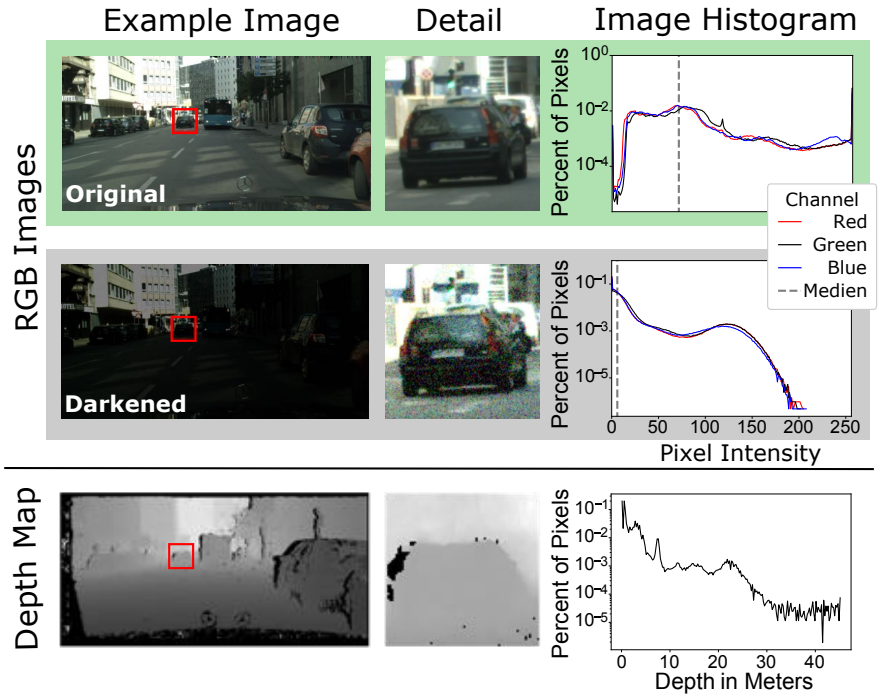


Figure 3.1: Comparing an RGB image with good exposure (first row) and an RGB image darkened with a physic-based model to imitate low illumination (second row) reveals the data challenges for semantic segmentation in dark environments. The detail, in the second column, shows an enlarged crop of the red bounding box on the example image. For the darkened image, we have brightened the detail to reveal the increased noise. In addition to noise, pixel intensities in the darkened image are skewed towards zero. The median pixel value changes from 72, in the original, to 7 in the darkened image. The increased noise and decreased dynamic range make RGB semantic segmentation networks more prone to error. The depth map, shown in the third row, has fewer semantic features, but is invariant to illumination, providing a stable modality in dark environments.

show that in low-light conditions the gains are more consequential, and therefore, utilizing depth sensors should be strongly considered.

RGB-D sensors are increasingly ubiquitous. In the past few years, commercial products such as phone and laptop cameras, gaming systems, cars, and robots have used RGB-D cameras to improve face recognition, computational photography, and navigation. These systems all operate in a variety of lighting conditions and could benefit from exploiting the depth channel in dark environments. However, few publicly available RGB-D datasets provide sufficient training data for end-to-end learning on new tasks, especially for low-light scenarios. We address this challenge by artificially darkening our training set using a physics-based darkening scheme.

We compare four different RGB-D semantic segmentation network architectures under different lighting conditions to explore which networks have the best performance, and conclude that simply concatenating the depth image as an additional channel to the RGB channels is simple and more accurate than other, more complex approaches. On a platform with a limited payload, such as an agile ground robot or quadrotor, there is a clear trade-off between weight and additional sensors. We found that while depth always assists with segmentation, its importance grows with increasingly limited illumination.

In this chapter, we show that fusing the camera and depth sensor measurements in an RGB-D image can improve semantic segmentation in dark environments far beyond its impact in bright environments. Furthermore, we demonstrate a particular approach to data augmentation and depth fusion that outperforms other state-of-the-art techniques. Finally, by using an early fusion approach, these improvements come at little computational cost.

3.1 Related Work

Standard CNN networks, such as ResNet and VGG16, trained on ImageNet are unstable for dark images [38]. These networks are frequently used as backbones for object detection and semantic segmentation. As a result, the downstream tasks are also prone to instability in low-light conditions. Fine-tuning on a dark dataset [39, 40] or on synthetically darkened images [41] can

improve performance and make these networks more robust to low-light scenes. Another approach to improve performance on dark images is to brighten the images. The process of brightening dark images is referred to as image enhancement. However, Xiao et al. [42] find that image enhancement does not produce better results than training directly on dark images. Work has also gone into developing object detection and segmentation networks that are robust in varying lighting conditions. Almost universally, these low-light networks use RGB input alone [42, 41, 43].

The largest barrier to developing low-light RGB-D segmentation methods is the lack of appropriate datasets. There exist low-light object detection datasets, such as See-in-the-Dark [44], ExDARK [39] and Night Owls [40], and night driving datasets which include segmentation labels such as Mapillary [45], Urban Night Driving Dataset[41], and Berkeley DeepDrive Dataset [46]. However, all of the above contain no depth measurements. Mapillary plans to release a depth dataset soon [47]. Synthia, a synthetic driving dataset, includes both depth and 8500 night images, but is too small for training purposes[48].

To generate a large enough RGB-D dataset under low-light conditions, we rely on physics-based synthetic darkening that accounts for the camera response, sensor and shot noise. Synthetic darkening by manipulating noise and response models has been previously used for estimating the noise characteristics of an image [49], and image enhancement [50, 51]. We use similar methods to augment our training set, and perform experiments under various "dark" conditions.

3.2 Method

3.2.1 Datasets

To train and test our networks, we used two semantic segmentation datasets, Cityscapes [16] and SceneNet RGB-D [52].

Cityscapes is an autonomous driving dataset collected during daylight with a stereo camera pair. Therefore, the dataset provides disparity images computed with semi-global matching [15] rather than depth captured with an active depth sensor. We use the train and validation splits

provided with 2975 fine-grained ground truth annotations training images and 500 validation images. We also use the 19,997 coarse-grained annotations provided as supplementary training data.

SceneNet RGB-D is an interior dataset created with a synthetic 3D renderer which automatically populates rooms with randomly placed objects. SceneNet's depth is directly computed from the camera position and the known location of the objects. Therefore, SceneNet represents ideal depth measurements. Consequently, SceneNet's depth is much cleaner and less noisy than Cityscapes'. Scenenet RGB-D has 5 million images however we used a more manageable subset of 292,803 images for training. From the provided 291,919 validation images, we randomly selected 500 images for validation and 500 images for testing.

3.2.2 Synthetically Darkening Images

Neither SceneNet RGB-D nor Cityscapes includes dark lighting conditions. Our solution is to synthetically darken images using a physics-based model.

For our physics-based model, we consider two aspects that change with illumination, camera response and noise. The camera response function varies from camera to camera and is usually measured directly under controlled lighting conditions [49]. We simplify this model to two components: a linear component due to fewer photons being captured by the CCD sensor, and a power-law component due to gamma-compression during image processing. In addition to modeling the camera response function, digital cameras have several sources of noise. We consider three: shot-noise, thermal noise (also called dark-current noise), and quantization noise. Shot-noise results from the Poisson process of photons arriving at the image sensor and is proportional to the number of photons or brightness of the image. Thermal noise results from the temperature of the image sensor, and is independent of the brightness of the image. Quantization noise results from the image-processing step of quantizing illuminance measurements into pixel values.

We approximate the observed (noisy) image $I = f(L + n_s + n_c) + n_q$, where f is a camera response function, $n_s \propto L$ is shot-noise (which is proportional to luminance L), n_c is thermal noise (which is independent of luminance), and n_q is quantization noise [49]. Following Li et al.

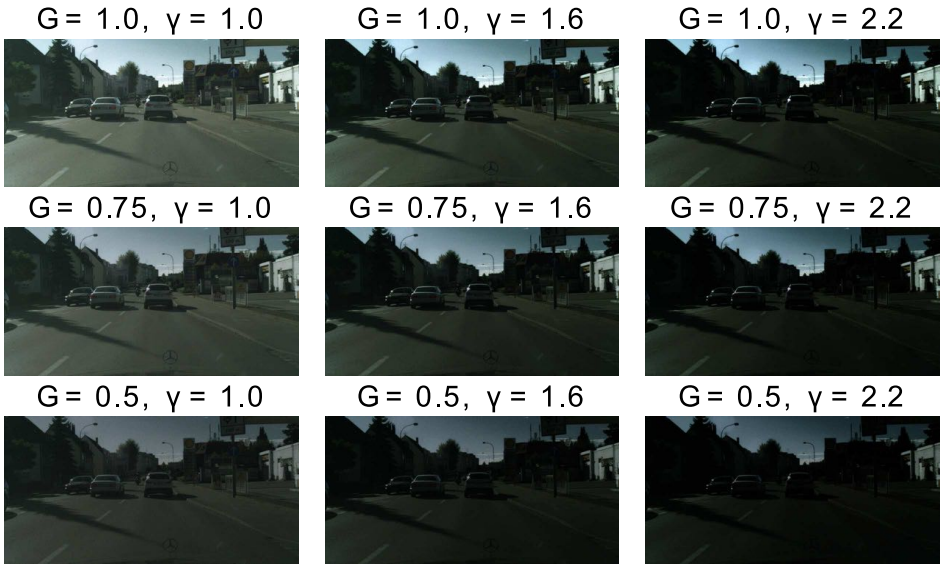


Figure 3.2: Grid shows a range of gain and gamma correction applied to an example image from Cityscapes. Top-right is the original image. The ranges represented, $G = [0.5, 1.0]$, $\gamma = [1.0, 2.2]$ are the same ranges used in training augmentation.

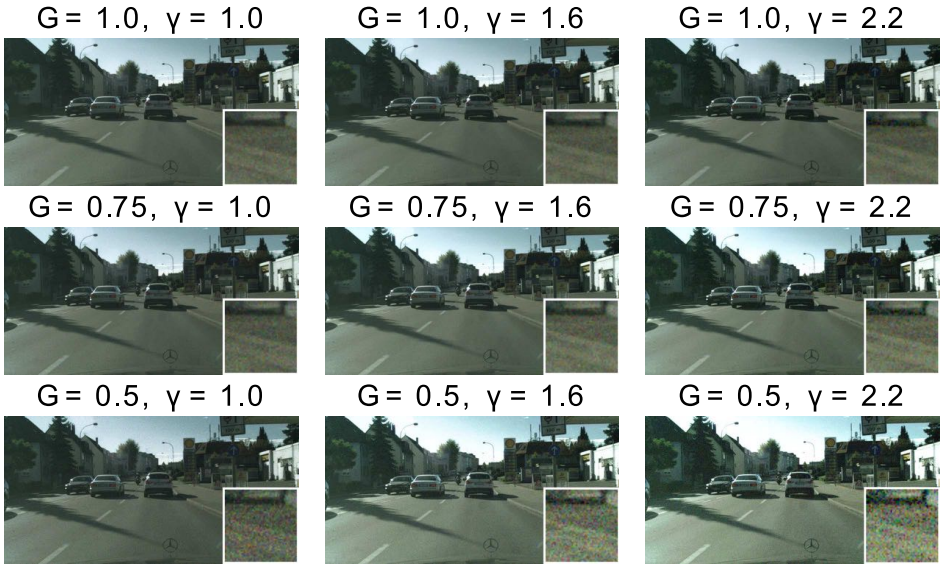
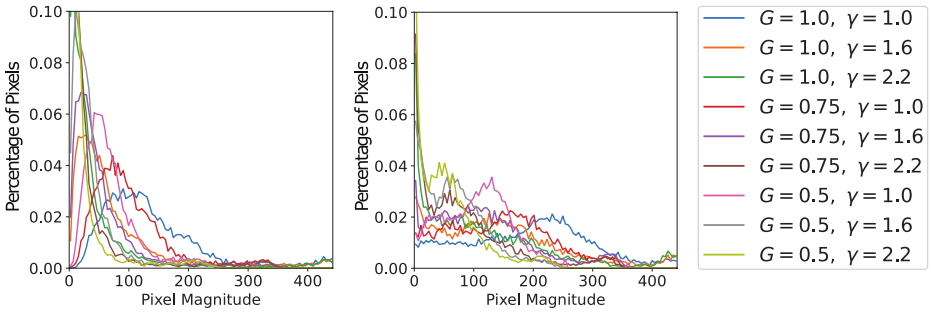


Figure 3.3: Grid shows re-brightened images from Figure 3.2 to illustrate the noise introduced by artificial darkening. Each image has an enlarged insert in the bottom right corner to show a close up of noise.



(a) Cityscapes (b) Scenenet (c) Legend

Figure 3.4: Histogram of RGB pixel magnitude under different darkening treatments. Percentage is estimated from a sample of 10,000 pixels. Clipping of the distribution toward zero pixel magnitude indicates underexposure leading to a loss of image information. The legend on the right is used for both plots.

[51], shot-noise is modeled as a Poisson distribution, and thermal noise is modeled as a Gaussian distribution with a standard deviation of $\sigma = 0.01$. Quantization noise is automatically produced by using a `uint8` image representation.

To model the camera response to lower-light environments, we adjust the gain and perform a gamma correction on our dataset. Adjusting the gain multiplies all pixels by the same value resulting in the uniform brightening or darkening of the image. This corresponds to fewer photons hitting the sensor. A gamma correction is a power operation which therefore affects dark and bright pixels differently. The gamma correction imitates the gamma compression applied to processed digital images. Figure 3.2 shows the same image under different gain and gamma correction. During training, each image is randomly augmented with a gain in the range $G = [0.5, 1.0]$ and a gamma correction in the range $\gamma = [1.0, 2.2]$.

These augmentations change the input images' characteristics in several ways compared to bright images. Firstly, the gain and gamma correction adjustments change the distribution of pixel values. Figure 3.4 shows the effect of the darkening treatment on the distribution of pixel magnitudes for both datasets. Secondly, the darker images have a much lower signal to noise ratio. Figure 3.3 shows the increasing effect of noise on darker images. Both of these effects will impact network performance.

3.2.3 Semantic Segmentation Networks

We train several different Deeplab v3+ networks [53] to demonstrate that including depth improves robustness across several different depth fusion methods. We choose Deeplab v3+ as it is one of the top performing semantic segmentation networks on the Cityscapes and COCO benchmark tasks. The Deeplab v3+ architecture is an encoder-decoder segmentation architecture which uses atrous convolutions to extract features at different scales. The encoder architecture can use different "backbones," or fully convolutional network architectures that extract features from the input images, such as ResNet [54] and Xception [55]. As we are using an early-fusion approach to incorporating RGB-D input, our modifications are to the backbones only, to allow them to accept RGB-D input.

We test different backbones to explore a range of possibilities for utilizing depth. In the original Deeplab v3+ work, the Xception backbone had the highest performance. Therefore, for the baseline RGB network, we use the Xception network, fine-tuned on the Scenenet or Cityscapes dataset. For RGB-D, we compared two different architectures, an early-fusion approach, a HHA mid-fusion network, and a Depth Aware Convolutional Network [32]. The early-fusion RGB-D network is an Xception network with the first layer modified to accept a 4-channel input. We will refer to this network from here on as the RGB-D network. The DepthAware Convolutional Network uses a custom convolutional layer which weights the convolutional kernel by the gradient of the depth image. The network's overall architecture is a ResNet [54], with the first convolution in each skip connection substituted for the depth-weighted convolution. We will refer to this network from here on as the DepthAware network.

All networks were trained from random initialization, with the exception of the RGB network, as no pre-trained weights are available for RGB-D networks with these architectures. Song et al. [31] shows that weights transferred from RGB training may actually be counterproductive when training RGB-D networks as the kernels learned on RGB channels and depth differ significantly. We experimented with transferring weights from one dataset to the other as a pre-training step, but saw no performance improvement. As a validation step, we also trained the RGB network from scratch and achieved similar performance. We concluded that the lack of pre-training was not negatively affecting our networks.

In principle, we could make similar modifications to any semantic segmentation network which had a separate "backbone," for example, Panoptic-DeepLab [56] or HRNet [37]. These networks released in the past several months achieve a performance of 85.1 and 85.4 mIoU on the Cityscapes benchmark while Deeplab v3+ achieves 82.1, but do not use the depth maps available. We would expect to see corresponding performance increases using these models together with our early fusion approach.

Table 3.1: Quantitative results on Scenenet without artificial darkening

Network	Acc	Class-m Acc	mIoU	FWIoU
Naive	0.50	0.04	0.04	0.25
RGB	**0.78**	0.55	0.42	**0.67**
DepthAware	0.77	0.46	0.36	0.64
RGB-D	**0.78**	**0.58**	**0.43**	0.66

3.3 Results

3.3.1 RGB-D Backbone Selection

Our first set of experiments focus on evaluating which network architecture has best performance in bright-lighting conditions. After identifying the best performing network, we train it with low-light augmentation.

The networks are evaluated on four quantitative measures, accuracy, class-averaged accuracy, mean intersection over union (mIoU), and frequency-weighted intersection over union (FWIoU). Accuracy (Acc) tends to be heavily weighted towards classes with large numbers of pixels. Class-averaged accuracy (Class-m Acc) corrects for this tendency by computing per-class accuracy and taking the mean. Both measures favor recall over precision. Intersection over union (IoU) is the ratio of true positive pixels to false positive and false negative pixels. FWIoU gives weights proportional to class representation, similar to accuracy, while mIoU averages the IoU score over classes similar to class-averaged accuracy. The IoU measures require both high precision and high recall for a high score.

Tables 3.1 and 3.2 show the results for all three networks and a naive assignment of the most frequent class for every pixel. For Scenenet, the RGB and RGB-D networks have comparable performance. For Cityscapes, RGB-D has the best performance. For both datasets, DepthAware struggles with the classes that appear less frequently in the dataset. Figure 3.5 shows example segmentation from Cityscapes for all four networks. Here we can see that qualitatively the RGB-D network has more detailed edges. Compare for example, the foliage in the first row. The DepthAware

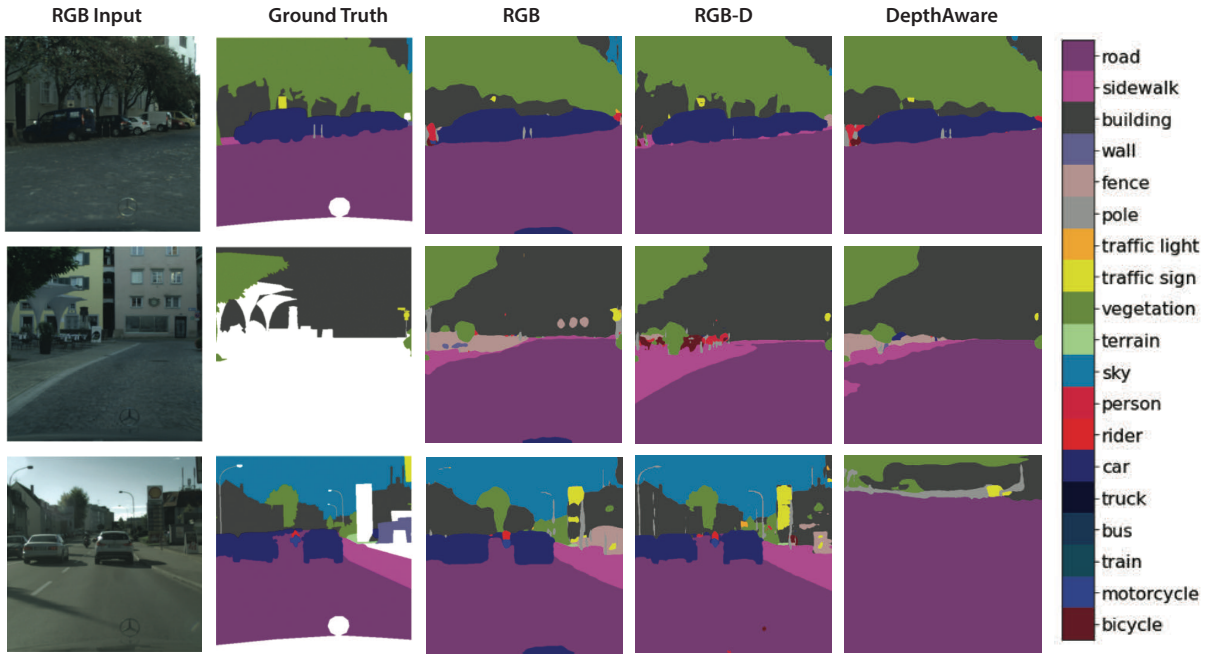


Figure 3.5: Examples of segmentation using different architectures. The first column is the input. The second column is the ground truth segmentation. The following columns are labeled with the network name. White pixels have no ground truth label and are masked during evaluation. The DepthAware network shows a failure mode in the third row.

Table 3.2: Quantitative results on Cityscapes without artificial darkening

Network	Acc	Class-m Acc	mIoU	FWIoU
Naive	0.43	0.02	0.02	0.18
RGB	0.93	0.74	0.58	0.88
DepthAware	0.92	0.70	0.56	0.86
RGB-D	**0.95**	**0.78**	**0.68**	**0.92**

network shows a typical failure mode in the third row, where the most frequent pixel class, road, is over-predicted.

Based on both quantitative and qualitative assessment, the RGB and RGB-D networks out-perform DepthAware. We therefore choose these architectures for our low-light experiments.

3.3.2 Feature Exploration

To explore how the RGB-D and RGB features differed, we used gradient-weighted class activation mapping (Grad-CAM) [8] to visualize class-discriminative features. To visualize the salient features for a class, c, the feature activations at a specified layer are weighted by the feature gradient at that layer calculated with respect to the score of the desired class, c. A ReLU is applied to the weighted value to emphasize features that have a positive influence on a prediction of c. The result is a heatmap with values between 0 and 1 where high values reflect areas with high salience to the class.

In Figure 3.6, we use Grad-CAM to visualize the last convolutional layer of the backbone network. This layer represents the highest level of semantics among the image features. Subsequent layers of Deeplab v3+ focus on upsampling for a higher resolution semantic segmentation. In fact, the Grad-CAM output is 13x13 due to the stride of the backbone network which leads to the aliasing visible in some of the activation heat maps. We compare the RGB and RGB-D models on four classes from the Cityscapes dataset; road, building, person, and car. The Grad-CAM heatmaps clearly show that the RGB and RGB-D models are learning different features for some classes. In particular, for the road and building classes, the RGB-D activations are high over the whole object

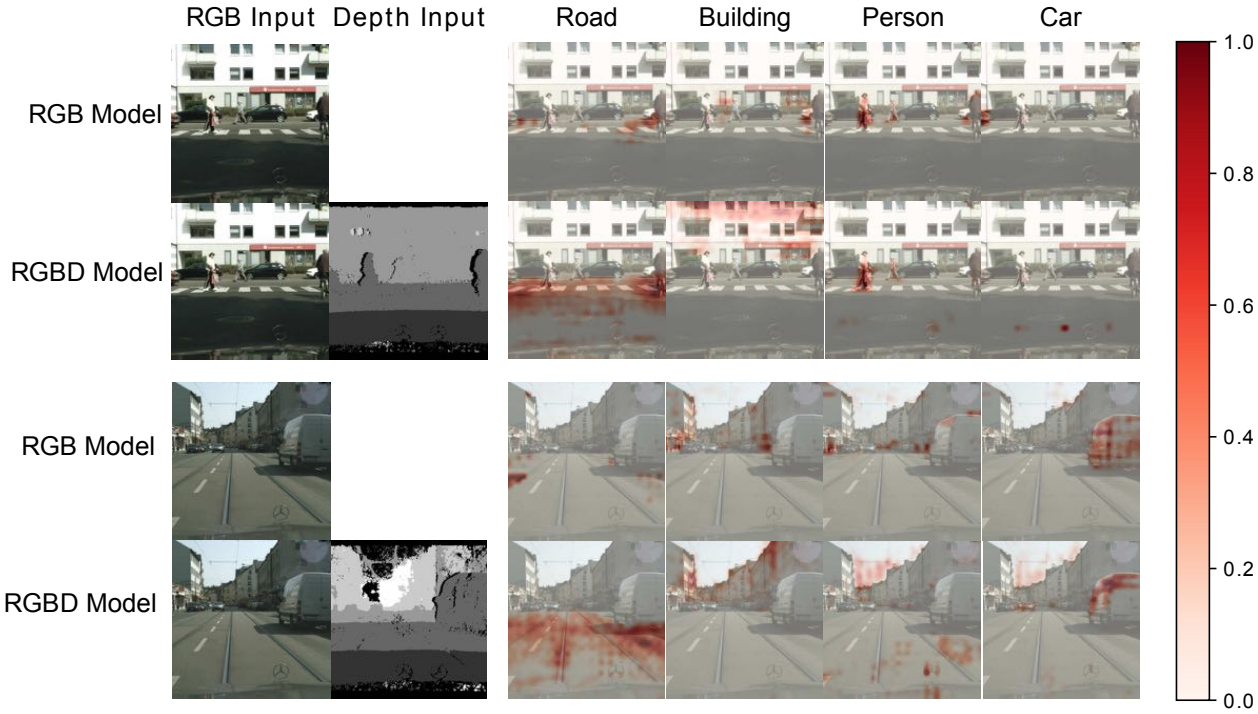


Figure 3.6: Activations after the backbone CNN for three object classes visualized using Grad-CAM [8]. The RGB and RGB-D models are learning different feature distributions. For example, the road activations show that the RGB-D model's feature applies to the whole road surface, while the RGB model

surface while the RGB activations are more concentrated on smaller areas such as the edge of the road or windows. In the second example, for the car class, the RGB model is active across the back of the car while the RGB-D model is only active on the edges of the car. This may be because the back of the car is a feature-less surface in the depth map, as it is all the same distance from the camera. The RGB-D model may be learning features that are discriminative in all channels, not just the color channels.

3.3.3 Low-light Results

For a quantitative analysis of the networks in different lighting conditions, we darken the whole test set and compute the mean IoU statistic for different gain and gamma values. In Figures 3.7 and 3.8, a representative image from the test set is shown together with a bar graph of the networks' mIoU scores. As gamma increases and the gain decreases, the RGB images become noisier and the performance of all networks decreases. The RGB network's performance decreases the most. The RGB-D networks have higher mIoU scores for all lighting conditions and the performance gap increases in low light. Additionally, training networks with low-light image augmentation improves their robustness to low gain, for both RGB and RGB-D networks. Consider the first column of Figure 3.7, the networks trained with low-light image augmentation perform better as gain decreases than those without.

Figure 3.10 shows a sensitivity study which investigates the contribution of each darkening variable to the decrease in performance. All other variables are held constant for each plot at their brightest value ($\gamma = 1.0,\ G = 1.0,\ \sigma = 0.0$). Here we see that low-light augmentation teaches the network to ignore Gaussian noise. Interestingly, networks trained on SceneNet are more robust to darkening than networks trained on Cityscapes. In addition to having less sensitivity as shown in Figure 3.10, Figure 3.8 shows that all networks trained on Scenenet stay well above the naive assignment threshold of 0.04 mIoU. For Cityscapes, at higher darkening levels, only the RGB-D networks are much better than naive assignment of 0.02 mIoU. This may be because the SceneNet images already contain noise and uneven lighting.

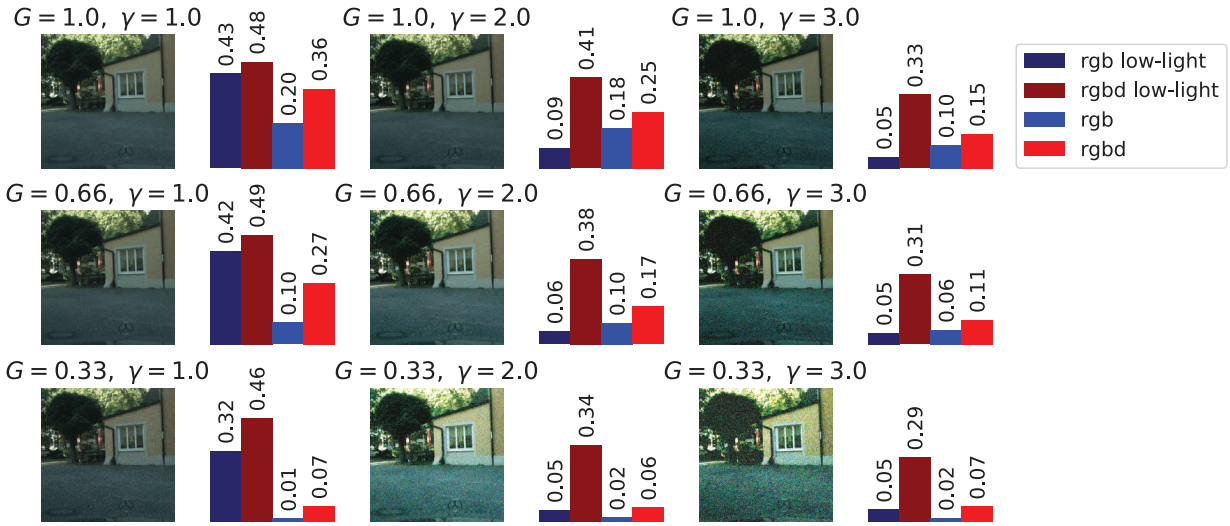


Figure 3.7: Bar graphs show the mIoU performance on the Cityscapes dataset with gamma-correction (γ) and gain (G) applied at test time. One example image is shown for each condition. All conditions have added Gaussian noise ($\sigma = 0.01$) and Poisson noise.

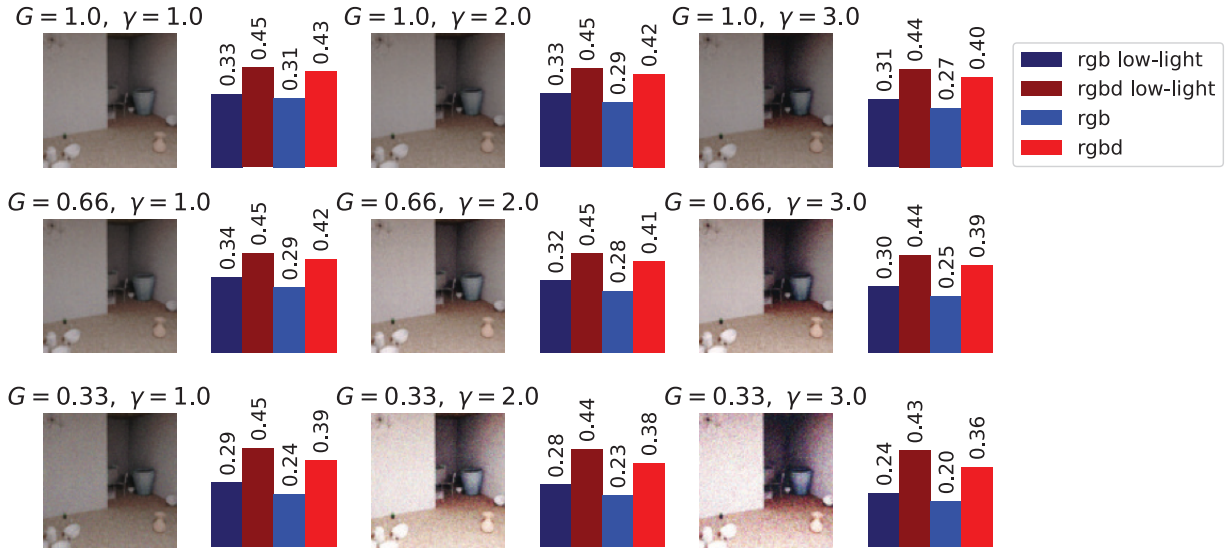


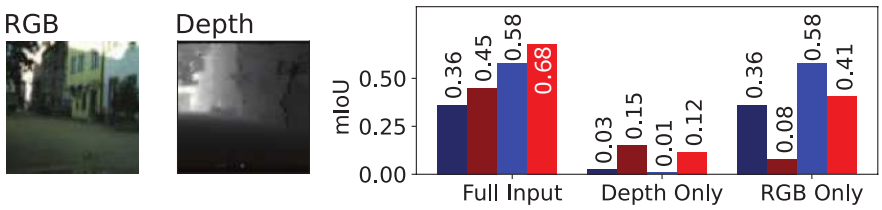
Figure 3.8: Bar graphs show the mIoU performance on the SceneNet dataset with gamma-correction (γ) and gain (G) applied at test time. One example image is shown for each condition. All conditions have added Gaussian noise ($\sigma = 0.01$) and Poisson noise.

A question that arises from the better performance of RGB-D is whether the network is relying on depth alone or whether the RGB channels are also contributing to its performance. Figure 3.9 shows how the performance of all networks is impacted by removing the RGB channels. On SceneNet RGB-D networks, removing the RGB channels has only a small impact, while removing the depth channel reduces the network performance to naive guessing. This suggests that the SceneNet RGB-D networks learn to depend heavily on the depth channel due to the noise in the dataset. In contrast, Cityscapes networks show large losses across the board when the depth channel is omitted. Interestingly, the RGB-D low-light network also shows large losses when the RGB channels are omitted. This suggests that the RGB-D low-light Cityscapes network is truly fusing the information from all four channels.

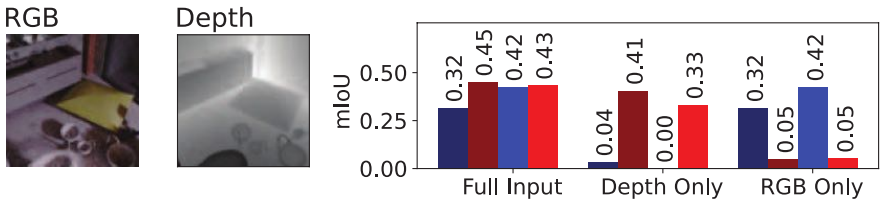
3.4 Conclusion

In dark environments, semantic segmentation greatly benefits from active depth sensors. Our experiments show the sensitivity of semantic segmentation networks to noise typical in low light conditions. Using RGB-D input counter-acts this instability as the network learns to rely more heavily on the depth channel. We have shown that the gap between RGB and RGB-D semantic segmentation performance increases as illumination decreases. By using early fusion architecture, this performance increase comes at little computational cost. These findings inform design decisions for platforms with limited payloads, such as quadcopters. Including a depth sensor on a platform operating in dark environments is worth serious consideration. Conversely, in bright environments, the gap between RGB and RGB-D performance is small, and a depth sensor can be omitted without a large performance loss.

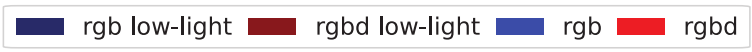
While this work focuses on semantic segmentation, the training methodology can be expanded to other tasks such as fast, light-weight object detectors like YOLO [57]. This is useful for agile platforms with more limited computational resources where object detection, not full segmentation, is desired. Such a system is critical to our DARPA Subterranean Challenge “artifact detection” system which needs to perform real-time object detection in dark underground environments as



(a) Cityscapes

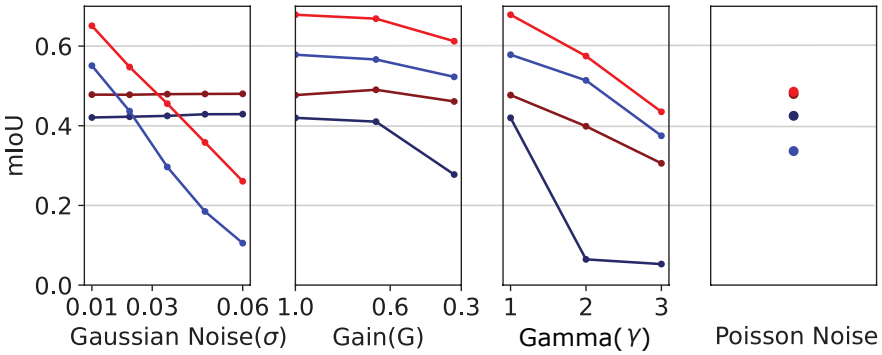


(b) Scenenet

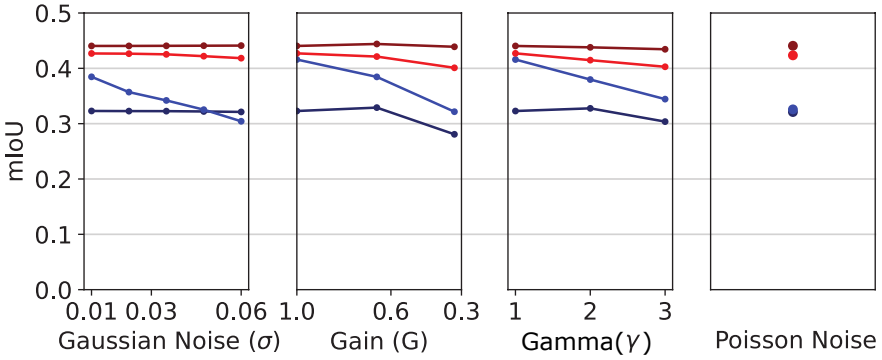


(c) Legend

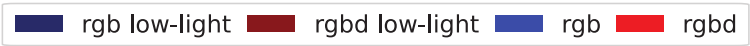
Figure 3.9: In an input ablation study, we compare network mIoU performance on full RGB-D input to input containing only a subset of channels. For each condition, the ablated channels are set to zero. The same legend is used for both datasets.



(a) Cityscapes



(b) Scenenet



(c) Legend

Figure 3.10: We run a sensitivity analysis where one darkening variable is perturbed at a time. Networks trained with low-light augmentation are invariant to Gaussian noise. For all other variables, darker values decrease network performance. The same legend is used for both datasets.

part of a search-and-rescue mission and is the topic of the next Chapter.

Chapter 4

RGB-D Artifact Detection for the Subterranean Challenge

In DARPA's subterranean challenge, teams compete to autonomously navigate a poorly lit underground environment and identify artifacts, such as a fire extinguisher, hand drill, backpack, and survivor, that could be important assets or targets during a search-and-rescue mission. The teams largely rely on RGB object detectors, such as YOLO [57], to identify these artifacts [58, 59, 60, 61, 62]. However, as described in the previous chapter, RGB images are typically degraded in low light environments. In under-exposed images, noise begins to dominate the signal. Moving platforms cannot increase exposure times due to motion blur. The degraded images, in turn, reduce object detector performance, which could lead the agent to overlook important artifacts including survivors during its mission. We propose an RGB-D object detection approach, based on lidar, to improve performance in dark subterranean environments.

In contrast to cameras, lidar does not have decreased performance in low-light. Because the lidar provides its own light source in the form of a laser, ambient illumination is unneeded. We therefore use lidar in combination with the RGB camera to improve object detection in dark environments.

4.1 SubT Artifact Detection

Artifact detection is the main evaluation criteria of the subterranean challenge. Teams must build platforms to autonomously map, plan, and navigate their way through the subterranean environment, but the ultimate goal is to find artifacts. Teams are ranked by the number of artifacts

(a) Backpack (b) Fire Extinguisher (c) Drill (d) Helmet (e) Rope

(f) Survivor (g) Vent (h) Cell (i) Cube

Figure 4.1: This figure shows all final event artifacts which can be localized using a visual sensor. Artifacts 4.1f-4.1i have additional detectable signals including heat(4.1f and 4.1g), Bluetooth(4.1h and 4.1i), and WiFi(4.1h). Not pictured is the CO_2 gas artifact which cannot be detected visually [9].

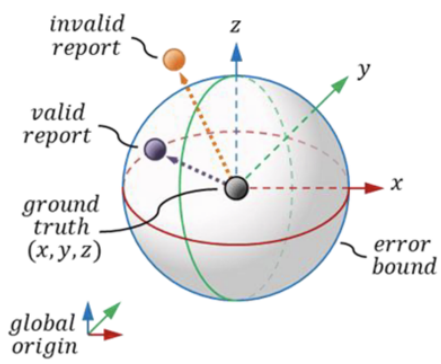


Figure 4.2: Artifact detection is scored on thresholded euclidean distance. The reported position of the artifact localization point must lie within 5 meters of the ground truth position to be considered a valid report [10].

which they locate in sixty minutes. Artifact detections are considered valid when the reported position is within 5 meters of the ground truth position (See Figure 4.2). In the final event, there are two rounds, the preliminary round which contains 20 artifacts and the prize round which contains 40 artifacts. To encourage precision, teams are limited to 25 and 45 scored reports respectively. There are 10 different artifact types, 9 of which can be detected visually (Shown in Figure 4.1). [9, 10]

In a typical object detection benchmark, for each object class, there are many different variations of that object. For example, for the dog class, there would be examples of many dog breeds. In contrast, the artifact detection challenge is specific to a single model of object, for example a Black and Decker GC960 Cordless Drill [9].

Usually, training an object detector on such a small set of objects would be an easy goal. In the subterranean challenge, detection is hampered by the challenging visual conditions. Courses include areas that are unlit or poorly lit and areas that have fog or smoke [9, 10]. The poor lighting conditions contribute to low contrast, high noise images. Increasing the exposure time increases motion blur. The platform used by the MARBLE team does have on-board lighting in the form of two 5W, 600 lumen, LED headlights. While the headlights are necessary in unlit environments, they contribute their own image processing challenges, including highly directional lighting, strong shadows, and sometimes overexposing close pixels.

The subterranean challenge provides an sandbox to compare the choices of several teams when it comes to artifact detection in a perceptually degraded environment. In previous competitions (Urban, Tunnel, and Cave), most teams relied on RGB sensing. The MARBLE [58], CERBERUS [59], COSTAR [60, 61] and CTU-CRAS-NORLAB [62] teams used YOLO as their object detector. Depending on the implementation, inference ranged from around 2.5Hz [62, 59] to 220 Hz [61]. MARBLE used Nvidia TensorRT to optimize the YOLOv3 architecture and achieved 60Hz inference [58]. The Explorer team chose to use MobileNet v1 because it was supported by the TensorFlow Object Detection API and YOLO was not. Optimization with the OpenVINO and Nvidia TensorRT libraries allowed for inference at an average of 60Hz of an Intel NUC [63]. The PLUTO team chose to use ERFNet, a semantic segmentation architecture, which operated at 5Hz [64]. All teams

aggregate detections over multiple frames to improve precision, using methods such as clustering [58, 64], Kalman filters [62, 61], and Bayesian filters [59]. In addition to the artifact aggregation, several teams relied on a human-in-the-loop to filter false reports and improve precision [62, 64].

Each team collected training data to fine-tune their networks on subterranean environments. CTU-CRAS-NORLAB gives minimal detail, saying only that they used "imagery gathered in other mines" [62], but other teams describe using multiple RGB sensors including cellphones and Stereolabs Zed Minicameras [64], a variety of lighting conditions [61, 59], and different environments including mines, caves, basements [64] and machine shops [59]. The Explorer team also used a RealSense active IR depth camera and a thermal camera to capture multimodal training data [63]. The resulting training sets range between 1000 [61] and 9712 examples [63]. To the best of our knowledge, none of these datasets is publicaly available.

To support the development of SubT SLAM systems, the Army Research Lab collected a public dataset. The SubT-Tunnel dataset was collected in the DARPA Tunnel Circuit using several robotic platforms including different sensor modalities [65]. Despite not being used by the authors for artifact detection, the ground truth positions of artifacts are labeled using professional surveying techniques, making it a unique, high quality test set for artifact detection. The SubT-Tunnel dataset includes a wide array of sensor measurements from different modalities, including two different lidar sensors, a stereo pair, a thermal camera, wheel odometry, IMU measurements, a CO_2 sensor, and measurements of 2.4GHz wireless signal strength. We use this dataset to evaluate our RGB-D artifact detector.

It is very challenging to evaluate the performance of the artifact detection models from previous competition circuits. The performance on the competition circuits depends on the planning and mapping capabilities of the platforms in addition to the object detection accuracy. For example, the COSTAR team was the top scoring team on the Urban competition. They found a total of 16 out of 40 artifacts. Of their missed detections, 20% were due to no robot traversing the area and 10% were due to the object not being in the field of view [61]. Ideally, we would want to eliminate these sources of error by using a standard dataset to compare all the artifact detection systems.

Several metrics can be used to measure the performance of an artifact detector using a dataset, instead of the in situ scoring used during competitions. Mean Average Precision (mAP) is a popular scoring metric for 2D bounding box detection and can be used to compare bounding box predictions. Relative Pose Error (RPE) is useful for scoring the predicted 3D pose of the artifact. However, teams that report mAP [63, 59] or RPE [64] scores do so on their own proprietary datasets. A public benchmark is needed for artifact detection to directly compare model results. Hence, our interest in the SubT-Tunnel dataset.

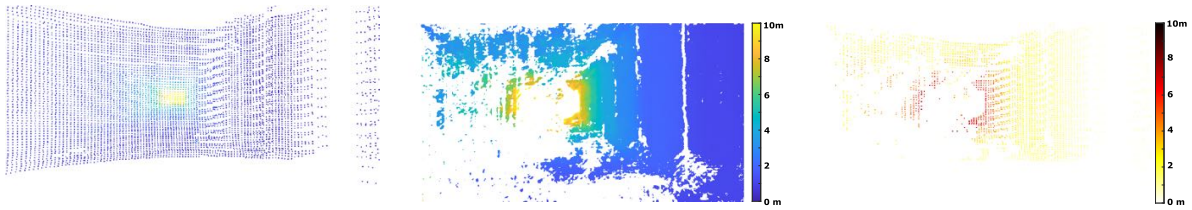
Other interesting multimodal methods that emerged from the subterranean challenge include, multisensor Lidar [66], RGB-thermal object detection [67], visual-inertial and thermal-inertial priors for lidar SLAM [68], and visual inertial SLAM for subterranean environments [69].

4.2 Method

4.2.1 Model

We use a YOLOv5 model [70] for object detection. YOLOv5 is a family of fast object detection models natively implemented in PyTorch. The models are based on the single pass YOLO object detection architecture introduced by Joseph Redmond [57], but with state of the art improvements including mosaic training augmentation [71], CSP backbone [72], and a PAN detection neck [71]. These improvements make it one of the fastest and most accurate single pass object detectors available off the shelf. The biggest difference between members of the family is model size, with the number of parameters ranging from 12.7 million in the smallest model to 141.8 million in the largest. Corresponding inference times range from 250 to 14 Hz.

We modify the YOLO models to accept RGB-D four channel input. This simple modification consists of changing the first convolution from a three channel convolution to a four channel convolution. We considered alternate RGB-D encoding schemes including HHA and Depth Aware Convolutions. HHA would double the number of parameters in the backbone, slowing inference. Depth Aware Convolutions did not perform well in our previous experiments in low-light environments



(a) Lidar (b) Stereo-Depth (c) Difference

Figure 4.3: An example of the difference between lidar and stereo-depth from the SubT-Tunnel dataset. Color indicates depth in meters. Even for pixels where overlapping measurements are available, the two methods of measuring depth have a 1.75 meter mean absolute difference. Lidar and difference pixels are enlarged for visualization.

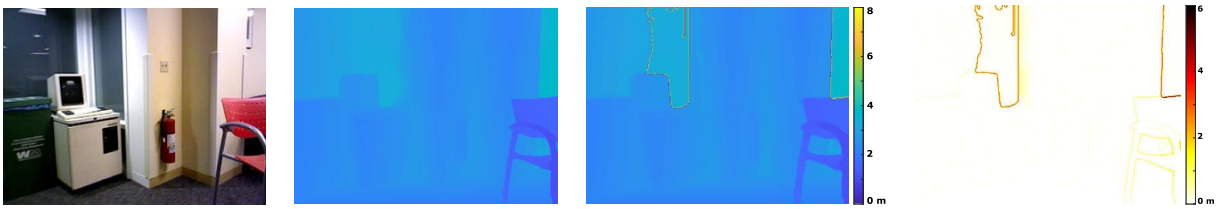
(See Chapter 3).

4.2.2 Data

The only object detection training datasets with both RGB images and lidar that we are aware of target autonomous driving applications, for example KITTI and nuScenes. Unfortunately, these datasets do not contain the object categories of interest for SubT. To learn these objects, we use a combination of RGB-D datasets and synthetically generated environments. For testing, we use the SubT-Tunnel dataset.

4.2.2.1 RGB-D Datasets

Datasets recorded with RGB-D cameras or stereo-pairs can be artifically manipulated to resemble lidar scans by subsampling the depth channel to imitate lidar sparcity. We sample 5% of pixels randomly to imitate the sparcity we observe on the Ouster lidar and on other lidar datasets such as KITTI. The lidar measurements are more precise than the stereo image depth measurements, meaning that simulating lidar with stereo-depth subsampling will be more noisy than lidar measurements. We measure a mean absolute difference between stereo and lidar-based depth measurements of 1.75 m on the SubT-Tunnel dataset sr_B_route2 run. An example of lidar and stereo-depth captured at the same location is shown in Figure 4.3. Larger errors are observed on more distant objects.



(a) RGB (b) Depth (c) IP-Basic (d) Absolute Difference

Figure 4.4: An example of SUNRGBD data. The depth channel is captured with a structured light sensor, in this case, an Xtion Pro. The IP-Basic depth completion is based on a 5% random sample of the depth channel. Error in IP-Basic is most apparent at the edges of objects.

For training, we use the SUNRGBD dataset which has 562 images of SubT relevant objects. SUNRGBD images are captured using commercial depth cameras such as the Xtion Pro and Kinect in interior spaces such as offices and homes. These spaces differ considerably from mines and tunnels, but it is one of the few RGB-D datasets which contains annotation for SubT relevant objects such as backpacks and helmets. An example of a SUNRGBD image and depth pair with simulated lidar-completed depth is shown in Figure 4.4. As SUNRGBD is captured in well-lit environments, we also use the dark augmentation described in the previous chapter to artificially darken the images.

4.2.2.2 Synthetic data

For synthetic data, we use the environment simulator and artifact models provided for the virtual track of the SubT challenge [73]. The simulator models an Ouster lidar as well as a front and back facing cameras.

To generate bounding box ground truth for the artifacts, we project a sample of 1200 to 2600 points from the artifact's surface to the camera's image plane. We use the maximal extents of this point cloud to define the artifact's bounding box. The number of points chosen depends on the size of the object model.

The simulator does not model the camera's exposure and thus has no camera blur. To imitate real world motion blur, we average every two consecutive images. The images are captured at 10 Hz, so this corresponds to a shutter speed of 1/5 s.

Dataset	Method	RMSE	MAE
SubT-Tunnel	Gaussian	2.50	1.92
	Bilateral	2.33	1.75
SUNRGBD	Gaussian	1.29	0.44
	Bilateral	0.89	0.24

Table 4.1: Evaluation of depth completion error using IP-Basic with different smoothing methods. RMSE is Root Mean Squared Error. MAE is Mean Absolute Error. The error metrics treat the stereo-depth (SubT-Tunnel) or depth measured using a structured light depth sensor (SUNRGBD) as the ground truth. The unit used is meters.

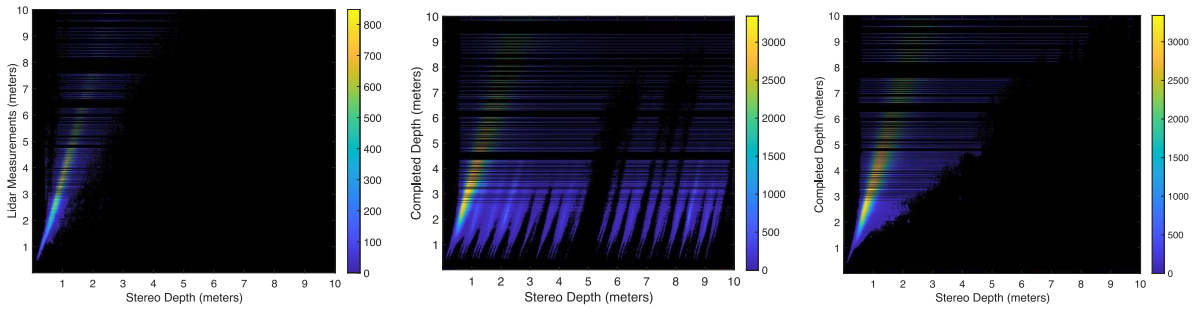
4.2.2.3 Lidar to Depth Map

The lidar measurements must be transformed to a dense depth map aligned with the camera image to be used as input to a conventional CNN object detector. We align the lidar and RGB camera using calibration parameters supplied by the simulation environment or by the dataset. For the SubT-Tunnel dataset, the lidar field of view overlaps 78% with the RGB camera. For the simulation environment, the lidar field of view overlaps 52% with the RGB camera. We crop the images to the intersection of the two sensors.

We project the lidar point cloud to the image plane, producing a sparse depth image. The value of each pixel is the distance between the point and the image plane. We use a depth completion method, IP-Basic [74], to densify the lidar image to the same resolution as the RGB image. IP-Basic consists of a series of morphological operations which interpolate between measured lidar points. No training or parameterization is needed, which makes it ideal for unusual domains like subterranean environments. It can be used off-the-shelf with no fine-tuning necessary.

Table 4.1 describes the error of IP-Basic when compared to stereo depth measurements or structured light depth measurements. It should be noted that stereo depth measurements in low-light environments are very noisy and a poor ground truth approximation. Figure 4.5c plots a confusion matrix showing stereo depth measurements versus lidar depth measurements for paired pixels across the dataset. We observe that stereo depth systematically underestimates depth values compared to lidar, so results in Table 4.1 should be weighted accordingly.

IP-Basic provides a few options for filtering operations including a choice between Gaussian



(a) Sensor Comparison (b) Gaussian (c) Bilateral

Figure 4.5: These three plots show confusion matrices for various depth completion techniques. Each (x, y) pixel represents the frequency that depth x was predicted to be depth y. Figure 4.5a shows the confusion between stereo and lidar depth measurements. We see that stereo measurements compressed range of depth compared to lidar. We expect to see that Figures 4.5b and 4.5c follow a similar distribution, but denser. Instead, we see that Figure 4.5b has many artifacts.

and bilateral smoothing. In the original paper, the authors find that that Gaussian smoothing produces the lowest RMSE, but in our experiments bilateral smoothing produced the lowest RMSE and MAE error for both datasets. Additionally, Figure 4.5b shows that Gaussian smoothing results in many completion artifacts. Ku et al. note that the bilateral smoothing is better at preserving local structure which may account for these artifacts. However, bilateral smoothing is slower than Gaussian [74].

4.3 Conclusion

In the past, our SubT platform has struggled to detect artifacts using a RGB YOLO approach. We hope to overcome the low-light visual noise by using lidar as an additional input. Multimodal RGB-Lidar input comes with its own challenges including re-projecting the lidar's 3D point cloud onto the camera's image plane, so that there is a correspondence between pixels and depth measurements. The lidar depth map must also be densified, so that it can be used as input to a CNN, but once these pre-processing steps have been performed, the object detection network architecture requires minimal modification.

This chapter extends Chapter 3's conclusions about semantic segmentation to object detection and demonstrates a field-ready RGB-D system for artifact detection. The successful application of

RGB-D features to two different computer vision tasks suggests that RGB-D features generalize well. In Chapter 6, I take generalization a step further to show how RGB-D features can support semantic understanding for a multimodal computer vision and natural language processing task, referring expression understanding, but first, what is referring expression understanding?

Chapter 5

An Introduction to Referring Expressions

When a speaker wishes to identify an object, they use a noun phrase called a referring expression. For example, "the coffee cup in the sink", "she", or even "the Museum of Contemporary Art" would be referring expressions. Referring expressions can also identify groups of objects as a collective reference, so "the large stack of books" is also a referring expression. The key property of a referring expression is that it distinguishes the referent from other objects [75].

As you can imagine, referring expressions show up everywhere. When not discussing generalities, e.g. "a cat can look at a king", discourse usually includes a referring expression or two. They are considered a key part of natural language generation and comprehension. To generate an utterance, the speaker must choose appropriate referring expressions. To understand what an utterance is about, the referring expressions must be resolved.

Consider Human Robot Interaction (HRI) as a motivating example of natural language generation and comprehension. To understand an instruction like, "Please water the flowers in the kitchen", a robot must be able to interpret the referring expression "the flowers in the kitchen". If the robot is confused, it must generate a different discriminating referring expression to ask for clarification, "Do you mean the flowers **on the counter?**" Turning referring expressions in instructions into an actionable plan is covered extensively in HRI research [76, 77] as is discriminative question generation [78, 79, 80, 81].

Referring expressions can also be used to learn about the environment. If the robot was asked to assist by fetching an unfamiliar tool, it could narrow down the possibilities by asking for

clarification as above, and then take a further step of building a model of the tool for future use based on the information that it has learned and observed during the process [82, 83].

The broad applicability of referring expressions means that there are threads of referring expressions research in many fields including natural language generation, computer vision, dialog systems, and human robot interaction. This means that different communities may be using different terminologies to describe the same problems. For example, throughout this thesis, I will term the subject of a referring expression, the referent, a term coming from linguistics literature [75]. However, in related research, the same object might be called the target, possibly coming from information retrieval, or the figure object, from Gestalt Psychology [84]. For other diverse terminology, I have provided a glossary (See Glossary) to explain equivalent or similar usage in different sub-fields which will hopefully assist the reader in their own exploration into related topics.

This chapter is a literature review of data-driven referring expression generation and comprehension, using images as the shared context between speaker and listener in which the referring expressions should be grounded. In Section 5.1, I introduce the role of referring expressions in natural language generation and discuss early work in the field of referring expression generation. In Section 5.2, I discuss the development of datasets for training and evaluating referring expression generation systems. Section 5.3 dives into the models used by such systems and the difficulty of evaluating referring expression generation. Section 5.4 addresses the inverse problem of referring expression comprehension.

5.1 Early Referring Expressions Work

Referring expression generation and comprehension has been a research topic since the early days of natural language processing. Gatt et al. [85] consider referring expression generation to be one of the main tasks in the natural language generation (NLG) pipeline. The others include content determination, text structuring, sentence aggregation, lexicalization, and linguistic realization. All of these tasks together allow a system to construct novel text from some underlying non-linguistic data representation. In the case of referring expressions, the underlying knowledge are the properties

that distinguish the referent from other objects.

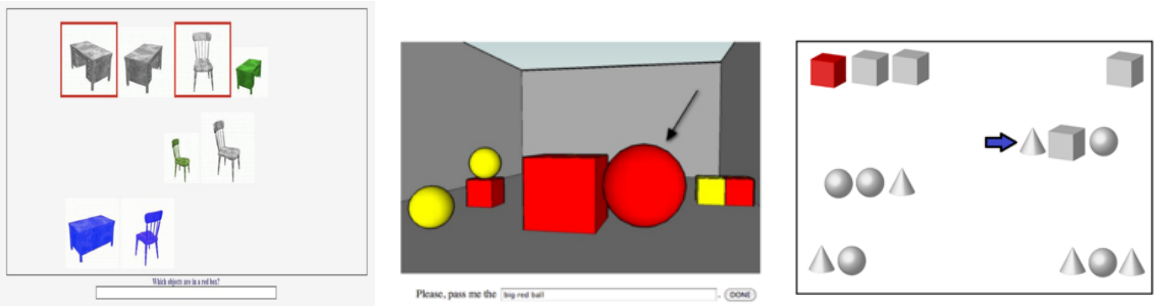
Several good surveys cover referring expression generation in depth. Krahmer and Van Deemter [86] focuses on referring expression generation specifically, while Gatt and Krahmer [85] surveys natural language generation more generally with a detailed section on referring expression generation.

An important initial step in referring expression generation research is defining what makes a well-formed referring expression. Most early work draws on the Gricean Conversational Maxims, a set of rules formulated by Paul Grice in 1975 to assist with viewing conversation through the lens of formal logic [87].

Grice lays out four categories of maxims, Quantity, Quality, Relation and Manner. The maxims of quantity state that a speaker should state as much as they need to in order to be informative, but no more. Essentially, be brief and to the point. The maxims of quality state that a speaker should try to stick to the truth. Don't make assumptions and don't say what is known to be false. The maxim of relation is the simplest; "Be relevant". Finally, the maxim of manner suggests style choices such as avoiding ambiguity and obscure reference, and organizing information meaningfully.

Much of the early work discussed in Krahmer and Van Deemter's survey [86], uses Grice's Maxims to prioritize how attributes should be used to disambiguate between objects when composing a referring expression. For example, the Full Brevity algorithm [88, 89] prioritizes quantity, looking for the shortest distinguishing description, while the Incremental Algorithm [89], a greedy solution, prioritizes manner and avoids more obscure attributes by choosing certain attribute categories above others. For example, size might be considered more informative than color.

The limitation of these approaches is that they assume that all possible descriptions can be enumerated and ranked. However, enumeration is costly. In the worst case, the Full Brevity algorithm might examine all possible combinations of attributes, and there are $n!$ possible orderings for the Incremental Algorithm's attribute hierarchy [1]. Usually only one is selected, but by what criteria? Creating the hierarchy is based on assumptions about human preference. In many open-



(a) TUNA [91] (b) GRE3D7 [90] (c) Stars2 [92]

Figure 5.1: Synthetic referring expressions dataset collection interfaces. In 5.1a, the annotator is asked to describe the objects in the red boxes. In 5.1b, the annotator is asked to describe the object pointed to by the black arrow, and in 5.1c, the annotator is asked to describe the object pointed to by the blue arrow.

world settings, it is not even possible to enumerate all the attributes and relationships that could be used to describe an object. Additionally, Grice's Maxim of quantity is frequently violated in practice. Viethen et al. [90] show that speakers will choose to use spatial information when other object attributes are sufficiently discriminative 13% of the time.

While treating the Gricean Maxims like rules of formal logic is appealing for designing algorithms, in the case of language, rules are usually more like guidelines. In 2007, Gatt et al. suggested that more systematic evaluation of referring expressions algorithms was necessary and introduced the first referring expression dataset [1]. Since then, data-driven models have emerged as the preferred choice for referring expression generation and comprehension.

5.2 Referring Expressions Datasets

To collect a referring expressions dataset, the researcher must elicit discriminative descriptions from a group of human annotators. An image or set of images is usually provided to motivate the referring expression. For example, Figure 5.1 shows three different collection interfaces. Datasets fall into two large categories; those containing synthetic images and those containing photographs.

TYPE	COLOUR	ORIENTATION	SIZE
chair	blue	forward	large
sofa	red	backward	small
desk	green	leftward	
fan	grey	rightward	

Table 5.1: TUNA's attribute sets [1]

5.2.1 Synthetic datasets

Synthetic datasets use computer-generated images to elicit and study referring expressions. They have been in use in psychology [93], NLP [1, 90, 92], computer vision [94] and human robot interaction [95] for the past ten years. The benefit of using synthetic datasets is that it is possible to control for object type, attributes, and frequency.

TUNA [1], the first purpose-built referring expression dataset, constructs a grid of either portrait photos (People) or furniture illustrations and asks the annotators to describe one or two target objects in the grid (See Figure 5.1a). Each referring expression collected was further annotated with the attributes' type and value, so that discriminative attribute sets could be formally represented and compared to those generated by models.

The furniture illustrations have a simple set of attributes categories (type, color, orientation, and size) and a limited number of variations within each category (See Table 5.1). TUNA's People domain, on the other hand, affords a much wider range of attributes, as humans are very sensitive to subtle changes in appearance. Gatt et al. note that annotators used unexpected attributes, such as referring to a particular hairstyle, which were not part of the expected attribute set. Additionally some human attributes chosen, such as age, may belong to a continuous spectrum. Already at this early stage in data-driven referring expressions research, the TUNA's People domain identifies a problem with the discrete enumerative schemes of the Greedy and Incremental Algorithms.

The next stride in dataset annotation was the inclusion of relational descriptions that involve a landmark object, like "larger than" or "behind" in GRE3D3[96] and GRE3D7[90, 97]. To more naturally elicit relational descriptions, GRE3D3 (containing 3 objects per scene) and GRE3D7

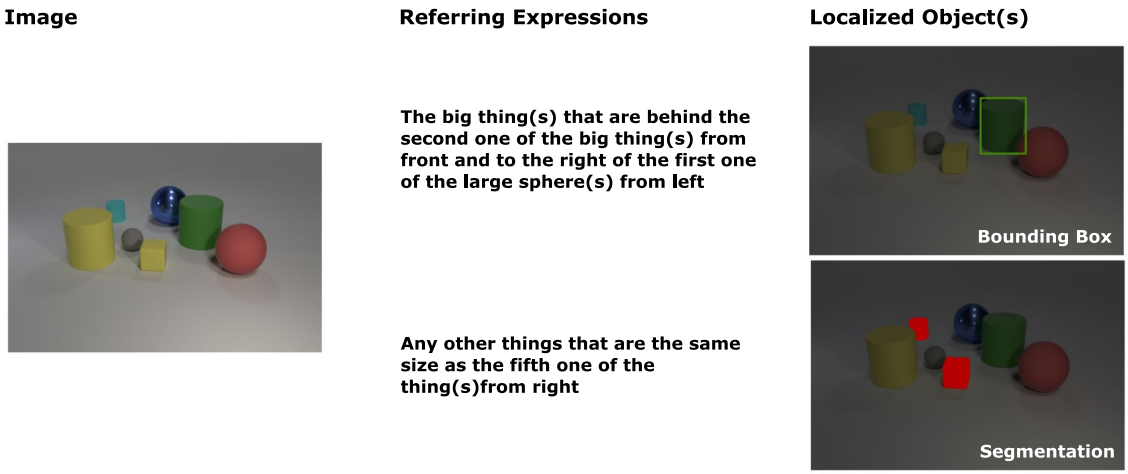


Figure 5.2: Examples from CLEVR-Ref+ dataset [11]

(containing 7 objects per scene), use a three-dimensional space containing simple colored shapes instead of an image grid. The spatial coordinate system more naturally elicits spatial descriptions like "on top" or "next to" from the annotators. However, a spatial description was **never** necessary to describe the target referent. There were always alternative discriminative descriptions using solely object attributes. This allowed Viethan et al. to measure annotator preference for attributes vs spatial descriptions.

Stars2 [92] was created to investigate the role of over-specification in referring expressions. In order to encourage over-specification, the authors of Stars2 include a large number of distractor objects, objects with similar characteristics to the referent, in each scene. Additionally, each scene includes an object with a highly salient property, for example brightly colored object (See Figure 5.1c). Including the colored object in the referring expression is not required to disambiguate the referent, but as color is a highly salient property, the hypothesis was that annotators would be more likely to over-specify a highly salient property, i.e. describe the color of the referent. Paraboni et al. were able to confirm this hypothesis [92], which suggests that referring expression generation should take the salience of distractor objects into account.

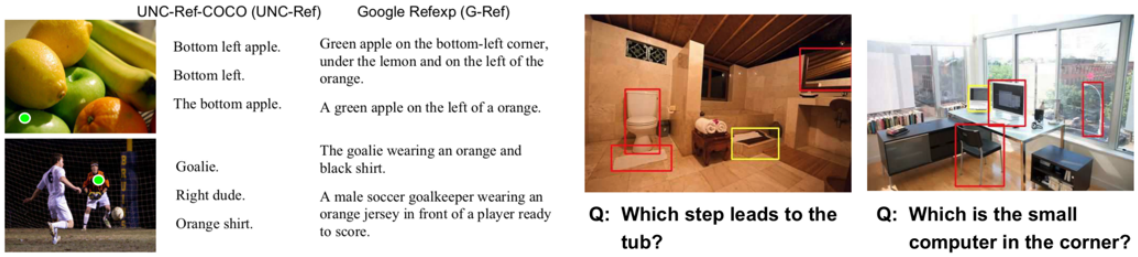
Distractor objects are important for studying dataset biases. In many computer vision tasks, a neural network can correctly guess the answer using data correlations within the dataset which are

not generalizable to new datasets, for example, predicting "dog" when an animal appears on a grassy background. CLEVR [94] was created to diagnose bias in Visual Question Answering systems by creating a synthetic dataset with many distractor objects and statistically uniform class and attribute occurrence. Liu et al. [11] extended this dataset with automatically generated referring expressions for diagnosing dataset bias in referring expression comprehension, CLEVR-Ref+. Their referring expression generation method uses templated referring expressions, so the language of the referring expressions is less varied than in other datasets collected from human annotators. Additionally, the referring expressions in CLEVR-Ref+ are extremely difficult for humans to understand. Figure 5.2 shows two example referring expressions. Unfortunately, while the idea of a bias-free diagnostic dataset is a valuable one, CLEVR-Ref+ falls somewhat short.

Common characteristics among these synthetic datasets include (1) full annotation and (2) a limited domain. Scene annotation includes all the information used to generate the scene, including the location and full set of attributes for each object including distractors. Referring expression annotation includes canonical labels for all attribute values, and in the case of GRE3D, GRE3D7, and Stars2, relational descriptions between referent and landmark objects. These comprehensive annotations support quantitative evaluation of referring expression generation methods. However, the cost of this comprehensive annotation is a limited vocabulary of objects and attributes. Each dataset exists in a closed world and would be difficult to generalize to real human environments.

5.2.2 Realistic Datasets

More recently, interest in training with large scale photo datasets, such as MS-COCO[2], sparked the creation of several large scale realistic datasets for referring expressions including ReferIt [3] and Google RefExp [4]. Yu et al. [99] package these two datasets into an easy-to-use GitHub package and extend them with additional expressions. These datasets are much larger than the synthetic datasets, in terms of number of images and referring expressions, and also have much larger vocabularies (See table 5.2. The hope is that by eliciting referring expressions on photos of human environments, these datasets will enable general purpose referring expression understanding.



(a) ReferIt (UNC-Ref) [3] and Google RefExp [4] comparison. Different collection methodologies lead to different levels of verbosity. Figure from [4]

(b) Visual7W Pointing Questions [98]

Figure 5.3: Referring expressions and scene examples from realistic datasets

ReferIt [3] used a game to elicit the referring expressions. Two annotators play cooperatively. One is assigned the task of describing the target referent. The other needs to correctly click on the referent based on the description provided by the first annotator. If the expression is not discriminative enough, the second annotator will not be able to correctly identify the referent. The less time the referring expression generation and comprehension cycle takes, the more points the annotators receive. The timed nature of the game encourages minimal expressions.

Not all collection methodologies result in minimal expressions. Google RefExp[4] paid annotators to provide referring expressions without any gamification. They collected referring expressions that were on average 4.8 words longer than those in ReferIt. Additionally, Google RefExp's annotators used a larger vocabulary, 4849 vs 2890 words used in more than 3 unique referring expressions. Examples of the different styles of referring expression are shown in Figure 5.3a [4]. The differing lengths of the elicited expressions suggests that style preferences for referring expressions are task dependent, another interesting result when considering Gricean Maxims.

The visual question answering dataset, Visual7W, also includes a referring expression comprehension task, which Zhu et al. [98] call "pointing" questions. This task pairs discriminative questions with several choices of bounding boxes. The answer must select the correct bounding box to answer the question. See Figure 5.3b for two examples. Each question includes a referring expression, although the form is slightly different than ReferIt and Google RefExp.

ReferIt, Google RefExp and Visual7W all use MS-COCO [2] as their source of images. MS-

Dataset	#Images	#REs	Vocab	Mean Length
ReferIt [99]	19,994	142,209	10,341	3.50
Google [4]	25,799	95,010	2,890	8.41
Visual7W [98]	47,300	327,939	-	6.9

Table 5.2: A comparison of three popular referring expressions datasets using the MS-COCO [2] images.

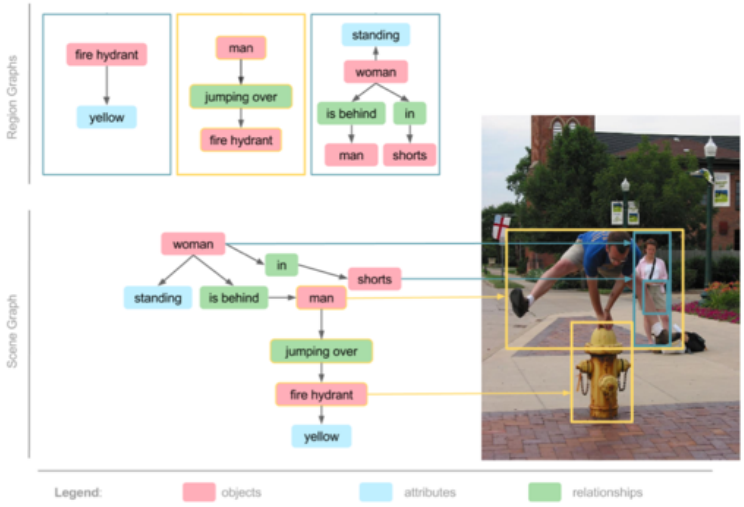
COCO provides 80 object categories and instance segmentation of 330,000 images collected from Flickr. Table 5.2 shows the number of referring expressions included in each of the referring expressions datasets. These are all sufficiently large to train neural networks for both generation and comprehension tasks.

There are several other realistic datasets which can be used for referring expressions research, including Flickr30k[101], Visual Genome [100], VQS[103], and What are you talking about? [102]. These datasets include grounding for referring expressions in sentence context. For example a question in a visual question answering tasks, "How many magnets are on the bottom of the fridge" contains the object description "magnets. . . on the bottom of the fridge". These datasets provide bounding boxes for each entity mentioned, e.g. the magnets, the fridge. Figure 5.4 shows a few examples. These datasets usually do not describe themselves as referring expression datasets, using the terms visual grounding [98, 103, 100] or image-to-text co-reference [101, 102] instead.

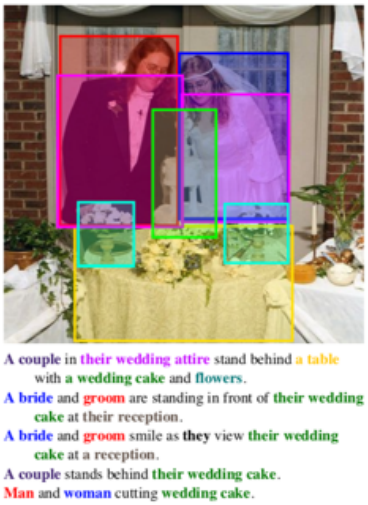
Other closely related data sets include visual dialog systems, like GuessWhat [80], where the goal is to generate a series of discriminative questions which zero in on one target object. HRI has also produced a few navigation data sets [104, 105] which use referring expressions to direct a robot to a goal.

5.3 Data-driven Referring Expression Generation

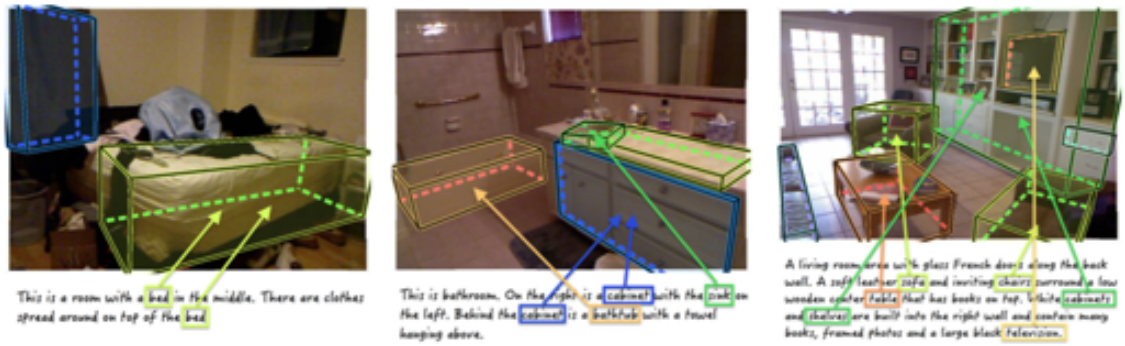
The creation of large realistic referring expressions datasets was driven by a desire to train deep neural networks. Neural networks had already been shown to help with language modeling and object detection, as well as joint visual-language tasks like image captioning. Many referring expression generation networks draw on these for-runners as architectural inspiration.



(a) Visual Genome [100]



(b) Flickr30k [101]



(c) What are you talking about? [102]

Figure 5.4: Datasets which provide referring expressions in sentence context

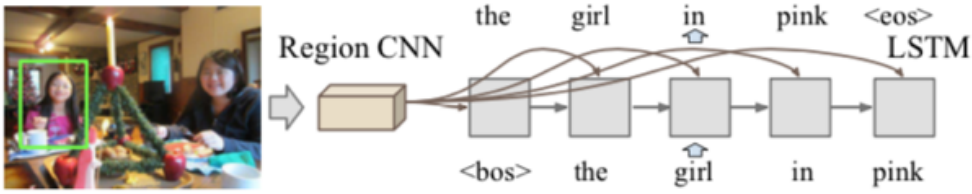


Figure 5.5: Mao et al.'s baseline referring generation architecture [4]

Most neural network approaches to referring expression generation are variations on the following structure. An image region is provided as the input indicating the desired referent. Hereafter, I will refer to this region as the target. Both the target and the whole image features are encoded into visual feature vectors using a convolutional neural network (CNN). The visual features are used to predict the referring expression using a sequence model like LSTM. Variations on this theme include extra input features, adding additional layers, or different training procedures.

Mao et al. [4] provides the baseline model. The network diagram is shown in Figure 5.5. A popular convolutional neural network (CNN), VGG-16, is used to represent the target and global image context. The VGG-16 network is pretrained on an object classification task to learn low level features like shapes and edges. The classification layer then is removed and the activations at the penultimate layer are used as the visual embeddings. The two visual representations, the target and the global image context, and the target region's coordinates are concatenated together. This input is provided to the LSTM at each time step together with a word embedding associated with the last predicted word. To start the sentence a beginning of sentence token ("<bos >") is used. The job of the LSTM is to predict the next word in the expression. During training, the loss function is the cross-entropy loss between the ground-truth referring expression associated with the desired target and the generated referring expression.

Mao et al.'s full model [4] adds a loss for distractor objects with the aim of making the generated expressions more discriminative. The new loss function minimizes the cross entropy loss for the desired target while maximizing the cross entropy loss for any distractor objects. To make this training procedure even more discriminative, objects of the same class as the desired target are chosen as distractors.

Instead of using a separate distractor loss, Yu et al. [99] train by generating descriptions of all objects in the scene simultaneously. To keep some distractor context, they also add another feature; the difference between the visual embedding of the target and the distractors of the same type. In an ablation study, the visual difference feature adds a 4-6% boost in referring expression comprehension.

The Speaker-Listener model [106] trains two models jointly, a generation model (speaker) and a listener model (listener). The Speaker is a straight-forward CNN-LSTM combination as in the baseline model. The listener transforms visual and referring expression representations into a joint space. Embedding Loss forces the two representations to be similar. The joint embedding space is used to rank referent proposals for the comprehension task (See Section 5.4). Since the same embeddings are used for the generative model, it will affect the generated referring expression. The hypothesis is that a visual embedding that is more similar to the referring expression representation may make it easier to reconstruct the correct referring expression.

Like the Speaker-Listener model, Liu et al. [107] train separate generation and comprehension branches of their network. They also train an intermediate attribute representation. The attribute learning requires choosing a set of attributes from the referring expressions, then training a multi-class classifier to predict whether an attribute will be used in the referring expression based on the referent region of the image. The attribute classifier is trained jointly with the rest of the model.

The attribute classification is used as input to LSTM in the generation model and as part of the feature in a comprehension model. In the comprehension model, another loss function is added to encourage the visual embedding and the attribute embedding to be similar to each other. The generation and comprehension models are trained jointly.

Both Hu et al. [108] and Nagaraja et al. [109] train on a loss function similar to the generative networks, but they are never evaluated as generative networks, only as comprehension models. Nevertheless, I include some details about their architecture here, as they are generative architectures.

Hu et al. [108] uses separate LSTMs to do joint embeddings of (1) the referent and preceding text and (2) the global image context and preceding text. The two joint embeddings are concatenated after the LSTM layer and the next word in the expression is predicted with a perceptron layer.

Nagaraja et al. [109] is one of the only generative models that attempts to model relationships between the referent and landmark objects. The representations of referent and landmark proposals are encoded as in the baseline network, but the loss function includes the current best-estimate of the landmark object.

	ReferIt		Google RefExp	
Method	ROUGE	METEOR	ROUGE	METEOR
Baseline [4, 107]	0.499	0.228	0.363	0.149
Mao et al. [4, 107]	0.497	0.228	0.354	0.144
Yu et al. [99]	0.533	0.249	0.370	0.151
Speaker-Listener [106]		**0.340**		0.159
Liu et al. [107]	**0.546**	0.258	**0.389**	**0.163**

Table 5.3: The results of all generation models evaluated on ReferIt[3] and Google RefExp[4] using the automated similarity metrics ROUGE[5] or METEOR[6].

5.3.1 Generation Evaluation

The gold standard for evaluating generated referring expressions is human judgment. Mao et al. [4] ask a group of annotators to rate whether the generated expression is more discriminative than the ground truth, similarly discriminative or less discriminative. The annotators are not told which expression is the ground truth and which is the generated expression. If the generated expression is at least as discriminative as the ground truth, it is rated correct. Another approach is to ask the annotators to click on the referent, in essence to perform the referring expression comprehension task for the generated expression. If the annotators can correctly identify the ground truth referent, the generated expression is rated correct. Unfortunately, both methods of collecting human judgment are expensive and time-consuming.

Automated metrics for assessing the quality of text are controversial. Many automated metrics exist, such as BLEU [110], ROUGE[5], CIDEr [111], and METEOR[6]. Traditionally used for measuring the quality of machine translation and image captioning, they can also be used for comparing the similarity of two referring expressions. Most of these metrics are sophisticated weighted overlap measures, similar to edit distance. Therefore, they have difficulty recognizing variation in descriptive text. BLEU and ROUGE in particular, have been criticized for not recognizing semantically equivalent sentences. However, ranking sentences using BLEU is shown to correlate well with human judgment [110], so while the sentence level ratings may vary in accuracy, a corpus level score can show trends in referring expression generation quality.

Mao et al. [4] proposes using an automated referring expression comprehension model to evalu-

(a) input image (b) object class segmentation of class ***people*** (c) object instance segmentation of class ***people*** (d) segmentation from expression *"people in blue coat"*

Figure 5.6: The differences between object segmentation, instance segmentation, and referring expression comprehension. Object and instance segmentation methods are only capable of distinguishing between a predefined vocabulary of object classes, while referring expressions include a much wider and more granular range of object descriptions. Figure from [12]

ate the quality of generated referring expressions. The hypothesis is that a well-generated expression will be more easily understood by the comprehension model, leading to a higher comprehension score. This might work for a fixed comprehension model which was tasked with comparing several different generation models. Unfortunately, Mao et al.'s comprehension and generation models are the same model (See Section 5.4 for details). I suspect that the comprehension model is simply reporting the model confidence in the generated referring expression. This is supported by the fact that Mao et al. report higher scores for generated expressions than for ground truth.

Table 5.3 compiles the comparable published results for the generation networks. The most widely available metrics are ROUGE and METEOR on the ReferIt and Google RefExp datasets. In each line of the table, the networks show incremental improvements on their predecessors. This suggests to me that the current generation approaches are slowly engineering small improvements, but that a dramatically different method may be needed to make a leap forward.

5.4 Data-driven Referring Expression Comprehension

Referring expression comprehension is the inverse task of referring expression generation. The goal is to take a scene and a referring expression as input and output the location of the referent. The location of the referent can be a segmentation or a bounding box.

Object segmentation and instance segmentation are like referring expression comprehension in that they produce a segmentation of objects in the image. Object segmentation labels all pixels with their object class membership. Instance segmentation additionally labels each instance of the class with a unique identifier. However, the objects that can be segmented are limited by a predefined vocabulary of object classes, whereas, referring expression comprehension allows for any discriminative object description. Hu et al. [12] provides a good figure illustrating the difference (See Figure 5.6).

There are two main approaches to referring expression comprehension, (A) ranking and (B) direct prediction.

5.4.1 Ranking Image Regions with Generative Models

In ranking, a set of candidate regions are compared and according to some criteria, one is selected as the referent. These systems are often not trained with the explicit goal of improving localization, but are usually trained as referring expression generation models. They use properties of the generative model such as vocabulary distributions or embedding space similarity to create a ranking metric for candidate regions.

The comprehension task is a useful side application for generative models. It can be implemented quite simply using existing parts of the model and reported performance is usually on par with the direct prediction models (See Section 5.4.3). Therefore, a ranking-based comprehension result adds to the usefulness of the referring expression generation model overall.

5.4.1.1 Generation Likelihood

One way of ranking regions is using the likelihood that a generative model would produce the referring expression given the region as input, i.e. $P(w_{1:n}|R, I)$, where $w_{1:n}$ is the referring expression, R is the image region, and I is the image.

From the generative model, we have $P(w_{1:n}|R, I)$, the probability of generating the referring expression. We can use Bayes' Rule to find the probability of the region given the referring expression

and the image.

$$P(R|w_{1:n}, I) = \frac{P(w_{1:n}|R, I)P(R|I)}{P(w_{1:n}|I)} \tag{5.1}$$

If we assume that the referring expression is equally likely to describe any of the regions that we are attempting to rank, we can assume a uniform prior for $P(R|I)$, which means,

$$\arg\max_R P(R|w_{1:n}, I) = \arg\max_R P(w_{1:n}|R, I) \tag{5.2}$$

Many neural network sequence models, like LSTM, contain a softmax classification layer that can be regarded as generating a probability distribution over the vocabulary at each step of the sequence inference. To find the likelihood of the sentence, the probabilities from each step should be summed,

$$P(w_{1:n}|R, I) = \prod_{i=0}^{n} P(w_i = w|w_{1:i-1}, R, I) \tag{5.3}$$

where w_i is a random variable representing the vocabulary at step i, and $w_{1:i-1}$ is the preceding sentence fragment [112]. Therefore, finding the best region corresponding to the referring expression, R^*, is a matter of finding the region which maximizes the product of conditional probabilities of each word in the sequence. For computational stability, this can also be expressed as the negative log likelihood.

$$R^* = \arg\min_R \sum_{i=0}^{n} -log(P(w_i = w|w_{1:i-1}, R, I)) \tag{5.4}$$

Similar conditional probability based ranking methods were introduced to this problem by Mao et al. [4] and Hu et al. [108]. Mao et al. were inspired by image retrieval methods which modeled the probability of a whole image given a caption [113, 112]. Hu et al. cites being inspired by the reformulation of object detection as object classification[114, 115]. Following these two papers, many researchers who designed a generative model used this technique to also evaluate their performance on comprehension [99, 109, 106, 116, 104].

Nagaraja et al. [109] also extend probability ranking to pairs of objects to capture spatial relationships.

$$R^* = \arg\max_{R \in C} \{ \max_{R_c \in C/R} P(R, R_c | w_{1:n}, I) \} \tag{5.5}$$

Here the idea is that the correct target region, R^*, will be most probable when the correct context region R_c is also selected. In addition to boosting target region prediction performance, using this formulation also identifies the context object, and can be used to ground additional objects mentioned in the referring expression.

While pairwise spatial relationships are common, referring expressions are not always limited to one context object. Nagaraja et al. [109] also propose a variation of Equation 5.5 using a "noisy OR" function to include context from more than one pair of regions.

$$R^* = \arg\max_{R \in C} \{ 1 - \prod_{R_c \in C/R} (1 - P(R, R_c | w_{1:n}, I)) \} \tag{5.6}$$

Since Equations 5.5 and 5.6 are pairwise comparisons, the computational complexity of checking all possibilities is $O(n^2)$ where n is the number of regions in the scene. Therefore, Shridhar et al. [116] propose a metric to limit comparisons to relevant pairs. They observe that relevant objects will have a higher $P(R|w_{1:n}, I)$ than irrelevant objects, and therefore, $P(R|w_{1:n}, I)$ can be used as a relevancy metric to cluster candidate regions and limit the number of comparisons.

5.4.1.2 Embedding space similarity

Another ranking approach, inspired by information retrieval, is to project the image regions and referring expressions into a joint embedding space and use a distance metric in that space to rank them. Regions and expressions that are closely related should project to closer points in the embedding space than those that are unrelated.

The referring expression generation models provide a starting point for the referring expression embedding. For LSTM-based sequence models, the hidden layer of the last time step is used. The

candidate image regions are typically encoded using all but the last layer of an object detection network such as VGG-16. Finally, the joint-embedding space is trained using a distance metric, such as euclidean distance [107] or cosine similarity [98], as a loss function.

Zhu et al. [98] train a network to minimize the dot product between (1) an embedding representing the referring expression and the whole scene image, and (2) an embedding representing the target region. The dot product represents the cosine distance between the two embeddings. Therefore, the network is essentially training the visual-text embedding (1) to be similar to the visual embedding (2). At test time, the candidate region which minimizes the dot product is considered to have the highest likelihood of being the target region.

Liu et al. [107] uses euclidean distance rather than cosine similarity as a loss function. Liu et al.'s network is competitive with other ranking networks on the ReferIt and Google RefExp datasets (See Table 5.4). There does not seem to be an advantage in training a distance metric compared to the likelihood approach. It requires extra training without significant comprehension gains.

5.4.2 Direct prediction

Like the generation networks, most approaches to direct prediction draw on existing network architectures. They are usually inspired by object or instance segmentation, while adding a text embedding of the referring expression to the feature space.

5.4.2.1 Attention-based models

GroundeR [117] uses an attention model to learn which bounding boxes are most important for reconstructing the input referring expression. Conceptually, the network takes an image, a referring expression, and several bounding box proposals as input. The embeddings of the image, referring expression, and bounding box proposals are similar to those described in Section 5.3. It is trained with an auto-associative task that must learn to reproduce the referring expression in the output layer. In the intermediate layers, the network must learn to distinguish the referent bounding box from distractors in order to accurately reproduce the input referring expression.

The attention layer is a two layer perceptron

$$\alpha_i = W_2\phi(W_h h + W_v v_i + b_1) + b_2 \tag{5.7}$$

where ϕ is the non-linearity, h is the text embedding, and v_i is the embedding of bounding box i. The attention α_i for each bounding box is used to weight the bounding box embedding before it is input to an LSTM to reconstruct the referring expression. At test time, the predicted referent is the box which received the most attention.

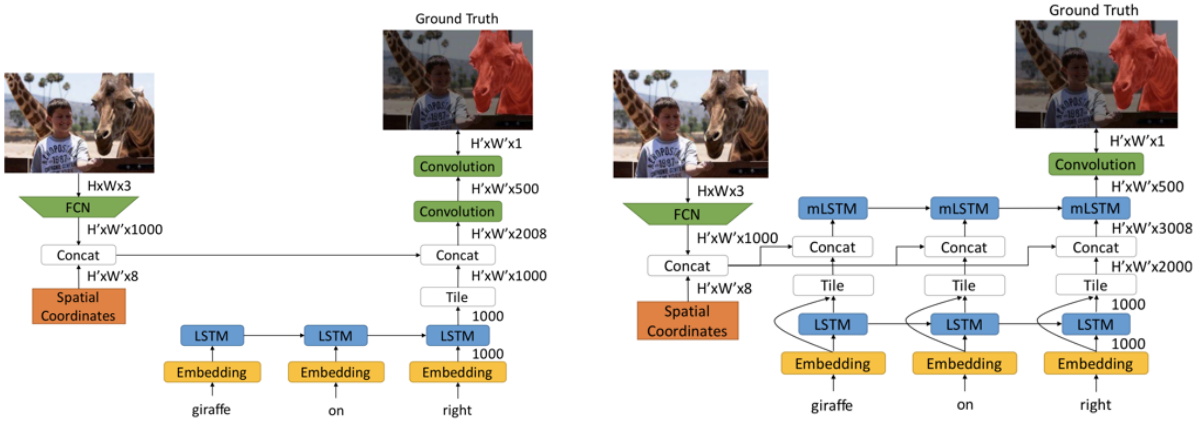
$$R^* = \arg\max_i \alpha_i \tag{5.8}$$

Fukui et al. [118] extend GroundeR with a more sophisticated method of combining the LSTM and CNN outputs. Instead of concatenation, they use compact bilinear pooling, an approximation of the outer product of two vectors. Compact bilinear pooling allows the network to model interactions between the visual and text embeddings. This approach does not lead to a significantly better result, but it is competitive with GroundeR (See Table 5.4) while requiring fewer parameters.

5.4.2.2 Segmentation Inspired Models

Hu et al. [12] use a well-established image segmentation architecture, FCN-32[119], and simply concatenate a sentence embedding to each image feature. FCN stands for Fully Convolutional Network. It calculates a low resolution feature map by convolving a CNN, such as VGG-16, over an arbitrarily sized image, $[W, H]$. The dimensions of the feature map are $[W/s, H/s, D]$, where s is the pixel stride of the final layer of the CNN and D is the output the dimensionality of the final layer. For example, with a pixel stride, $s = 32$, a $512x512$ image becomes a $16x16xD$ feature map. In order to turn this low-resolution visual feature map into a high-resolution object segmentation, a convolutional classifier classifies each pixel with an object label, and the labels are up-sampled with the inverse operation of the max-pooling that was applied during the convolutional step.

In the case of Hu et al., several other features are concatenated to the low-resolution visual



(a) Hu et al.'s segmentation model concatenates the visual and text features [12]

(b) Liu et al.'s segmentation model uses two LSTM layers, one to encode the text followed by one to encode a joint text-image representation [13]

Figure 5.7: Network diagrams illustrating the differences and similarities between [12] and [13]. Both use an image CNN to encode the input image and a LSTM to encode the text features. Both use final layers which include a convolutional classification layer to label each pixel followed by deconvolutional upsampling. Figure from [13]

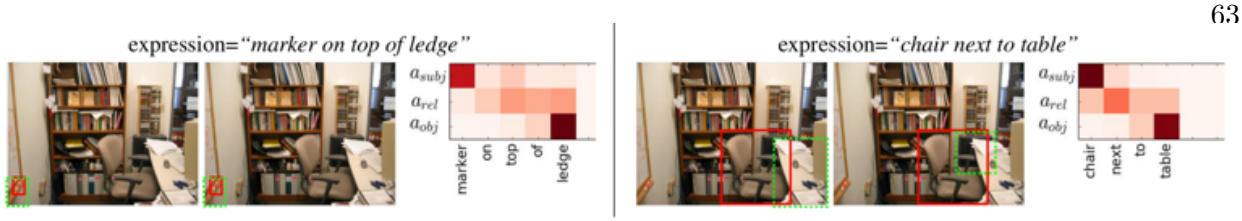


Figure 5.8: Results from Hu et al.'s CMN. Ground-truth is shown in the first column, followed by the predicted regions. The referent is in the red solid box and landmark is in the green dashed box. The third column visualizes attention weights where darker is higher attention. Figure from [14]

feature map before the classification layer. These include image coordinates and a LSTM embedding of the referring expression. The image coordinates assist with referring expressions containing spatial information like "on the left". Image coordinates are necessary because the final convolutional classification layers are spatially invariant and are unable to distinguish between features from the left and right of the image unless coordinates are themselves a part of the feature. The LSTM embedding is similar to those described in Section 5.3. Finally, instead of choosing an object label, the convolutional classification layer classifies each position in the feature map as part of the referent or background. The full architecture is illustrated in Figure 5.7a.

As with Rohrbach and Fukui, Hu et al.'s concatenation based segmentation was quickly followed by a more sophisticated method of combining features. Liu et al. [13] use a second LSTM to combine the features sequentially. Instead of first embedding the whole sentence and then concatenating the result with the feature map, the second LSTM (labeled as mLSTM in Figure 5.7b) receives the text embedding and the feature map as input at each step of the text embedding's computation. This additional layer results in a 10% reported improvement to Hu et al.'s network (See Table 5.4).

5.4.2.3 Modular Networks

Hu et al. [14] introduce a new network type called a Compositional Modular Network (CMN). The model learns relationship triples, i.e. (subject, relationship, object), where subject is the referent and object is the landmark object. Each component of the triple is selected using an attention layer.

In GroundeR[117] and Fukui et al. [118] the attention layer weighted bounding box proposals. In CMN, the attention layer weights the words of the referring expression. The weighted sentence embeddings are then input to either the localization module or the relationship module.

The localization module scores a single bounding box on how likely it is either the referent or the landmark object. The output of the attention layer is element-wise multiplied with a visual feature representing the proposed object region to produce a joint visual-text embedding. The score is a linear combination of the features of the joint embedding.

The relationship module scores a pair of bounding boxes for how likely the pair matches the representation of the relationship in the referring expression. The visual embeddings of the two regions are concatenated and fed through a perceptron layer to produce an embedding compatibly-sized for element-wise multiplication with the output of the attention layer. Again, the score is a linear combination of the features of the joint embedding.

In a fully supervised setting, the training loss of the CMN optimizes the pairwise score of the ground-truth referent and landmark objects. Since not all datasets provide ground-truth labels for landmark objects, Hu et al. also propose a weakly supervised training loss which maximizes the unary score of the ground-truth referent by taking the argmax of all possible landmark objects.

Figure 5.8 shows the success of CMN on predicting the relationship triples using the attention layers. Yu et al. [120], however, argue that referring expressions do not always have a (subject, relationship, object) structure. They propose instead a modular network which separates expressions which only reference the subject, e.g. "the pink donut", from those that include location, "donut on top", and those that include a relationship "donut on the plate".

Yu et al.'s modular network, MAttNet, uses a very similar attention mechanism to CMN, but makes several changes to the modules. An attribute prediction loss is added to the subject module. The relationship and location modules include context features from five surrounding objects in addition to the region currently being scored. For all modules, instead of learning a joint-embedding space, cosine similarity between the referring expression embedding and the visual region embedding is used for the scoring function.

	Method	p@1 (IOU=0.5) ReferIt	p@1 (IOU=0.5) Google	Mean IOU ReferIt	Mean IOU Google
Ranking	Mao et al. [4]	70.0	60.7		
	Hu et al. [108]	72.74			
	Yu et al. [99]	67.64	46.86		
	Speaker-Listener[106]	72.78	64.0		
	Nagaraja et al. [109]	**75.6**	**68.4**		
	Liu et al. [107]	72.08	52.35		
Prediction	GroundeR[117]	28.51	48.38		
	Fukui et al. [118]	28.91			
	Hu et al. [12]	34.02		48.03	
	Liu et al. [13]			58.73	34.52
	CMN[14]		**69.3**		
	MAttNet[120]	**79.55**	65.60	**62.37**	**48.61**

Table 5.4: The results of all comprehension models that evaluated on ReferIt [3] or Google RefExp [4] datasets.

MAttNet, while among the most competitive on the comprehension task, does not beat CMN's precision (See Table 5.4). Therefore, their argument that relationship triples are a worse representation of referring expressions needs more evaluation.

5.4.3 Comprehension Evaluation

There are two standard metrics for assessing the quality of segmentation and bounding box results; intersection over union (IOU) and precision at k (p@k).

IOU is the ratio of the intersection of the predicted bounding box or segmentation and the ground truth to the union of the same. IOU balances precision and recall of the predicted area. If the region predicted is too large, due to a high recall regime, the union of the two regions will be large, decreasing the score. If the intersection of the two regions is small, either because it is too conservative, due to a high precision regime, or because it predicts the wrong location, the score will likewise be lower.

The IOU enforces more precision on smaller objects. Imagine a coloring book with a few very small detailed areas versus a large object area. Coloring slightly outside the lines in the detailed

area would result in a larger union and decrease the IOU score. However, the same sloppiness on the border of the large object would have a much smaller proportional effect.

Mean IOU measurements were only published for the segmentation models[12, 13, 120]. All other models simply used IOU for a relevance threshold while calculating precision at k.

Precision at k ($p@k$) is a metric originating in information retrieval. In information retrieval, p@k measures the number of documents relevant to the search criteria in the set of k top retrievals. In the case of referring expression comprehension, we consider our search criteria to be the target object and the relevance of the result to be the IOU of the predicted object and the target object. For example, $p@1$ with $IOU = 0.5$ would measure the percentage of test set exemplars where there was at least a 0.5 IOU score between the predicted object and the ground truth object.

Table 5.4 shows all the models for which we have comparable comprehension evaluation scores. In general, progress is incremental. Most models are adding only a few additional components to their predecessor, such as a new way of composing the text and image feature or a few additional layers to the network.

The top models are the modular networks [14, 120] and Nagaraja et al. [109]. The modular networks are the most customized for the task. They may perform best because the most feature engineering went into them. They also include separate modules for encoding pairwise relationships. Nagaraja et al.'s [109] comprehension innovation is to use pairwise spatial relationships when ranking the proposed referents.

5.5 Conclusion

Referring expressions are still a challenge with state-of-the-art models performing at around 70% p@1 and around 60% mean IOU for comprehension. The importance of referring expressions suggests that more research should be pursued in this area. The most promising future research directions seem to involve modeling both referent and landmark objects and their relationships. Large improvements are seen in referring expression comprehension from modeling relationships[14, 120, 109].

Referring expression generation models that include relationships have been less well explored. Nagaraja et al. [109] is a generative model, but was never evaluated for its performance on referring expression generation. It would be interesting to measure the expressions it produces with ROUGE or METEOR and compare them to other generative models.

To evaluate referring expressions involving relationships, semantically aware similarity metrics should be explored. Using semantically invariant representations of referring expressions, like AMRs, could lead to better sentence level comparisons.

Chapter 6

A New Dataset for Spatial Referring Expressions

Imagine you are sending your robotic assistant to fetch the USB drive you forgot in your office. You tell the robot that "It's either on the desk next to the mouse, or it's in the top drawer on the right of the computer." Considering the clutter on your desk, this description is critical to successfully retrieve the USB drive.

Spatial descriptions, such as "next to the mouse", are a natural way of describing objects and are often preferred by speakers when differentiating between objects. Viethan et al. [90] show that speakers will choose to use spatial information when other object attributes are sufficiently discriminative 13% of the time, suggesting that these expressions provide helpful information even after an object has been sufficiently identified.

At the same time, spatial referring expressions are especially challenging to interpret because they require understanding additional contextual information. Appearance based descriptions like color, shape, or object class, require detecting the attributes of the target object alone. In contrast, spatial descriptions require understanding the relationship between the landmark object and the target object and are often perspective dependent.

Existing visual referring expressions datasets, such as ReferIt[3] (sometimes called RefCLEF due to annotating the ImageClef[122] dataset), RefCOCO [99], and Google RefExp[4](sometimes called RefCOCOg), include spatial relationships but are dominated by appearance-based descriptions.

Parts of this chapter appeared in Mauceri et al. [121] and the copyright is held by 2473-9944/19/$31.00 ©2019 IEEE. Research sponsored by DARPA Award: FA8750-18-2-0016. Some class statistics have been updated from the previously published version. Thank you to Tim O'Gorman for providing a list of spatial prepositions.

Additionally, they lack a depth information, which is desirable for robotics applications [123, 124]. This Chapter describes **Scene Understanding for Spatial Object localization** (SUN-SPOT), a new dataset which combines RGB-D images with spatial referring expressions. SUN-Spot contains 7,987 referring expressions with an unprecedented average of 2.6 spatial prepositions per expression. An example from the dataset is shown in Figure 6.1.

SUN-Spot's annotations are most similar to the Google RefExp dataset likely because we use a similar collection methodology. The main differences are our focus on spatial relationships between objects in cluttered interior scenes, and the composition of the images. The photos used by Google RefExp, ReferIt, and RefCOCO are gathered from photo collections on the web. SUN-Spot images are keyframes from a video stream and therefore more closely resemble the visual input of a mobile robot. Characteristics of the keyframes include bad lighting conditions, non-level camera frames, and a large amount of clutter. Google RefExp, ReferIt, and RefCOCO images are usually well lit, mostly level, and have a small number of highly salient objects. Finally, the most important difference is that the SUN-Spot images include a depth channel, while Google RefExp, ReferIt, and RefCOCO have RGB images only.

Depth has been shown to improve accuracy for scene understanding [119, 32, 125, 126] and manipulation [123, 124], but it requires additional hardware to collect compared to conventional images. Therefore, datasets based on RGB-D images are rare. A few simulated referring expressions datasets, such as GRE3D7 [90], include depth information, but are not photo-realistic. There are future plans to include depth observations in the photo-realistic Matterport3D Simulator which is used to generate images for the Room-to-Room navigation dataset. To our knowledge, our dataset is the only dataset which provides referring expression annotations for RGB-D photographs [104].

To create SUN-SPOT, we labeled a subset of 1948 images from the popular scene understanding dataset, SUN RGB-D [18]. SUN RGB-D provides images with a depth channel, object category labels, instance and semantic segmentation, and object pose annotations. Examples of the SUN RGB-D annotations and depth channel are shown in Figure 6.2.

The rich set of SUN RGB-D annotations combined with SUN-SPOT's spatial referring

expressions mimic real world HRI applications where robots are typically equipped with depth sensors and humans prefer to use spatial object descriptions. We see this dataset as providing a challenging new baseline for referring expression generation and comprehension which will support advancements in HRI.

6.1 SUN-SPOT

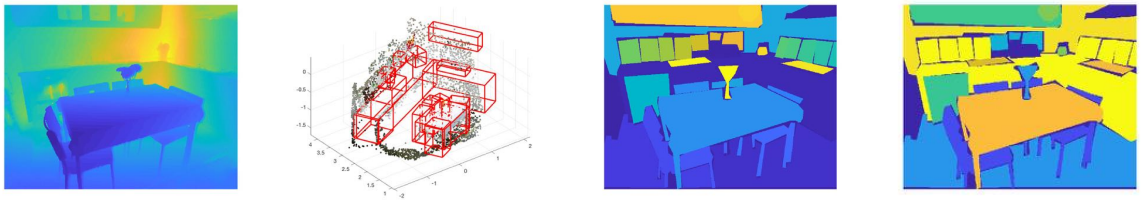
Our SUN-SPOT dataset extends the popular scene understanding dataset, SUN RGB-D [18]. SUN RGB-D contains 2d object segmentation and 3d object bounding boxes with orientation for over 10,000 RGB-D images of indoor scenes (See Figure 6.2). A subset of 1,449 images, originally the NYUv2 dataset [129], has been previously annotated with captions [102] and visual questions [130].

We annotated 1,948 images with referring expressions for 578 unique object classes. The images were selected with a focus on images containing two or more objects from the same object class, similar to the methodology of Google RefExp [4]. The need to discriminate between objects of the same class within the same scene forces the annotators to provide more detailed descriptions, and allows for meaningful error analysis on the final comprehension results. We can observe when the comprehension falls back on object class detection and when it successfully uses the spatial information in the referring expression. Unlike Google RefExp, we also include some images with only one instance of the object to achieve a somewhat balanced distribution of the object classes occurring in the SUN RGB-D dataset.

Our image selection process first computed the number of occurrences of each object class in each image. For each multiply-occurring object class, we then selected a random sample of 50 images containing at least 2 objects of that class. Some classes like "oven" never appear more than once in the same image. For these classes, we take a random sample of 10 images depicting these classes to avoid excluding object classes that appear in the SUN RGB-D dataset. For example, Figure 6.1 shows two labeled objects from the class "flowers" and one labeled object from the uncommon class "calendar." Duplicate objects are rare. Remarkably, only 6.5% of images contained more than two

- The calendar is hanging below the cupboards above the sink
- The calendar is over the sink.
- The flowers in the corner of the room, to the right of the silver plaque.
- The flowers are on the corner of the counters, to the left of the range.
- The flowers are on top of the white table in a clear vase.
- The red, white and yellow flowers sitting in the middle of the table.

Figure 6.1: An example image from the SUN-SPOT dataset including three objects with labeled referring expressions. We annotate objects which have multiple class instances in an image (i.e. the flowers in the green and yellow bounding boxes) to test the ability of referring expression systems to differentiate between objects. We also annotate classes which occur rarely (i.e. the calendar in the blue bounding box) to balance the frequency of object classes and test zero-shot learning. We collect multiple referring expressions for each annotated object. A selection of annotations for the three highlighted objects in this image are shown on the right.



(a) (b) (c) (d)

Figure 6.2: SUN RGB-D annotations for the image shown in Figure 6.1. From left to right, we show the depth map (blue is closer to the camera, yellow is farther), a 3D projection of the RGB-D image with overlaid 3D object bounding boxes, instance segmentation (each color labels an object instance), and semantic segmentation (each color labels an object class).

instances of the same object class.

6.1.1 Expression Collection

We collected referring expressions from workers on Amazon Mechanical Turk at a cost of around $0.15 per expression. The workers were shown an image with one object bounding box highlighted and were instructed to describe the location of that object relative to other objects in the scene. They were also instructed to avoid using image coordinates (e.g. "in the upper right corner"). Workers had the option to skip images that they found particularly challenging to describe.

A second group of Turkers were tasked with validating the referring expressions. They were shown the same image, without the bounding box highlight, together with the expression. We requested they click on the object described by the expression or to flag the description as problematic. The options for problematic flags included no fitting object being visible, the description being incomprehensible due to spelling or grammatical errors, or a custom option where they describe the problem in a text box.

Each expression's referent had to be correctly identified by at least one validator to be included in the data set. We also filtered out expressions which had a high ratio of misspelled words, did not include a location preposition, or used image coordinates.

We showed each object to at least 3 annotators in step one. However, because annotators had the option to skip images and other expressions were discarded after validation, the mean number of referring expressions per object is 2.46.

6.1.2 Landmark Object Annotation

After the initial collection, we determined that it would be useful to have coreference chains which associated the landmark object in the image with the tokens referencing the landmark object in the referring expressions. These annotations are inspired by coreference chains annotated in Flickr30k [101]. An example of an annotated coreference chain is shown in Figure 6.3. This annotation schema supports quantitative analysis of the ability of a referring expression system to

(a) Image (b) Landmark regions highlighted

Figure 6.3: Example of landmark annotations for the referring expression:

> **"The sofa is located near the center of the room, in front of a wall with hanging scrolls of Chinese lettering, facing away from the wall. The sofa has an end table with a white top attached to its (visually) right side."**

The candidate landmark phrases are highlighted in colors which correspond to the region highlight colors in 6.3b. This example contains a successful coreference ('a wall' and 'the wall'), a failed coreference ('chinese lettering', and 'hanging scrolls') and a reference to the target object which was not the subject of a sentence ('it').

ground landmark objects and also introduces the possibility of using landmark objects explicitly during training. For example, Bhattecharyya et al. developed a fine-grained multimodal retrieval system which is sensitive to entity roles using the coreference chains from Flickr30k [131].

To collect the coreference chains, we first used SpaCy [132], a Python NLP toolkit, to extract noun phrases from the referring expressions. We used the 2.3.1 "en_core_web_sm" model, which is a multi-task CNN trained on OntoNotes [133]. Its reported accuracy on part of speech tagging is 97.06 and on unlabeled dependency parsing is 91.66. We made the assumption that the subject of the referring expression would be the target object, and that any other nouns are landmark objects. This assumption sometimes fails. One common example is sentences that start "There is a ...", for example, "There is a sofa in the center of the room". Another failure case is shown in Figure 6.3 in coreference chain (5) where the target is referenced using a possessive pronoun. We therefore describe these noun phrases as candidate landmark phrases.

To link the candidate landmark phrases to their corresponding image regions, we use a web annotation interface similar to that used to validate the referring expressions. A group of Turkers were shown the image with the referring expression and a candidate landmark noun-phrase highlighted and underlined. The Turkers were asked to click on the highlighted and underlined object. We recorded their click location, and indexed the ground truth image labels from SUN RGB-D at that location to identify the landmark object. To ensure high quality landmark annotations, at least two Turkers had to click on the same image region. If three Turkers could not break a tie, the landmark phrase would remain un-annotated. As with the other tasks, the Turkers also had the option to skip any question they chose.

One benefit of using image regions to identify landmark objects, is that coreference decisions can automatically be collected from tokens that are assigned the same image region. For example, in Figure 6.3, the coreference between **'a wall'** and **'the wall'** was determined by annotators clicking the same image region for these two noun phrases. Additionally, target objects which were misidentified as landmark candidates using noun phrase parsing can now be linked to the target object region and removed from the landmark candidate list. Identifying coreference chains with

Dataset	Images	REs	Vocab	Classes	Length	LocPrep
SUN-SPOT	1948	7,990	2,690	578	14.04	2.60
ReferIt (RefCLEF)[3]	19,997	130,364	9,320	276	3.51	0.76
RefCOCO [99]	19,994	142,209	10,341	80	3.50	0.87
Google RefExp (RefCOCOg)[4]	25,799	95,010	2,890	80	8.41	1.23

Table 6.1: A comparison of SUN-SPOT to three popular referring expressions datasets. While SUN-SPOT is an order of magnitude smaller than these other datasets both in the number of images (Images), and the number of referring expressions (REs), it has a higher frequency of location prepositions (LocPrep), longer average expressions (Length), and more object class labels (Classes).

image regions sometimes fails when the landmark object is a collection. For example, in Figure 6.3, there are several wall hangings. The part-of-speech parser incorrectly split the noun phrase "hanging scrolls of Chinese lettering" into two candidate landmark phrases. If annotators for both phrases had selected the same image region, they would have been coreferenced. Because there were multiple wall hangings, they coincidentally choose different image regions for each phrase leading to a failed coreference.

In total, we collected annotations for 14,151 landmark objects and identified 1453 coreference chains where the same landmark was referenced multiple times in a single referring expression.

6.1.3 Dataset Statistics

Table 6.1 summarizes the size and complexity of the resulting dataset. It has relatively long descriptions compared to other referring expressions datasets and a larger average number of spatial prepositions per annotation.

The most similar dataset in terms of expression length and frequency of location prepositions is Google RefExp. This could stem in part from the similarity in our collection methodologies. We take a closer look at the distribution of location prepositions in Figure 6.5. To calculate the frequency of location prepositions, we use a vocabulary of common location prepositions and count token occurrences in the dataset. Since these prepositions may have multiple senses, this calculates an upper-bound on their usage as location prepositions. Eight of the ten most frequent location prepositions are shared by SUN-SPOT and Google RefExp. Figure 6.5 shows that several

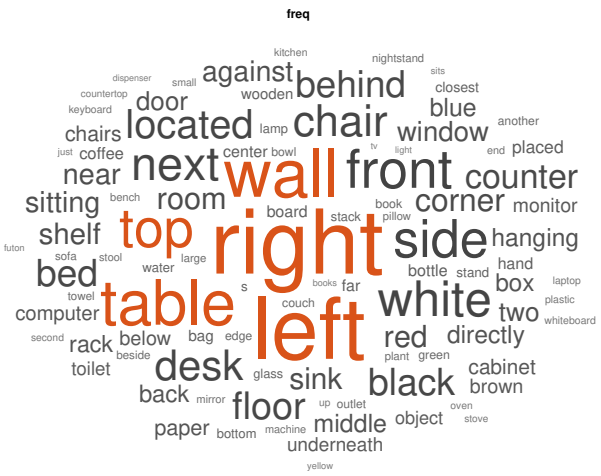


Figure 6.4: Word cloud of the most frequent vocabulary in our dataset. The vocabulary focuses on objects and locations.

prepositions that unambiguously refer to location, such as 'on,' 'left,' 'right,' 'front,' and 'top,' occur in greater proportion in SUN-SPOT.

6.2 Experiments

In order to compare the challenges of learning on SUN-SPOT to other referring expressions datasets, like Google RefExp, we ran several experiments using the same baseline referring expression generation network as Mao et al. [4]. Furthermore, to investigate the value of SUN-SPOT's depth channel, we modify the baseline slightly to accept depth input and compare to Google RefExp with synthetically generated depth. Our implementation and trained models are available as a Github repository[1] .

6.2.1 Models

Mao et al.'s [4] network consists of two main components, an image network to encode the image features, followed by an LSTM to generate text.

For the image portion of the network, a pre-trained VGG16 network [134] extracts an image embedding at the last 1000-dimensional layer of the network. The VGG16 network is used to embed

[1] `https://github.com/crmauceri/ReferringExpressions`

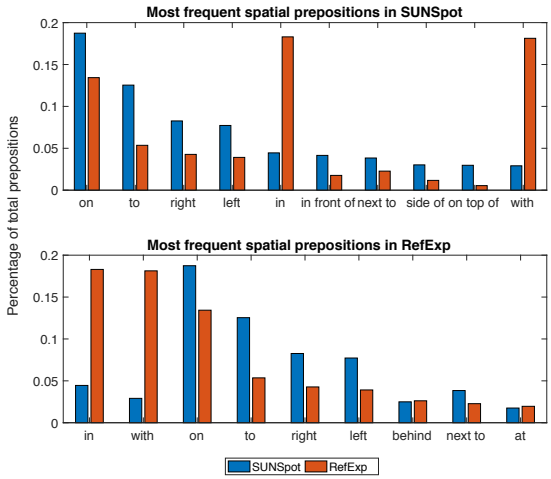


Figure 6.5: Relative frequency of the 10 most frequent location prepositions in Google RefExp and SUN-Spot

both the full image (used as global context) and a cropped image of the target object. Both images are scaled to 224×224, while leaving the aspect ratio the same. The remaining pixels are padded with the mean pixel value. Finally, a five-dimensional vector representing the position and relative size of the bounding box is appended to the two image embeddings resulting in a 2005-dimensional visual representation.

For the text portion of the referring expression input, 1024-dimensional word embeddings are randomly initialized for the vocabulary. A joint image-text embedding is produced by concatenating the word-embedding to the image-embedding resulting in a 3029 dimensional image-text representation.

The joint representation is then fed to an LSTM which is trained on a next token prediction task. The LSTM has a 1024 dimensional hidden layer, and is trained using cross-entropy loss. Cross entropy loss minimizes the negative log likelihood of generating the ground truth referring expression. For each word, w_i in the sequence, the loss is

$$J(w_i) = -\sum_{j=1}^{n} \mathbb{1}_{i \neq j} - log(p(w_j))$$

where n is the size of the vocabulary, $\mathbb{1}$ is an indicator function which is 1 when i is not equal to j and 0 otherwise, and $p(w_j)$ is the probability of generating vocabulary word w_j. We summed the

loss over all words in the sequence. While the formulation is slightly different, we believe summed Cross Entropy Loss is equivalent to Mao et al.'s "Maximum Likelihood Training" loss.

The architecture described above is a re-implementation of Mao et al.'s system. However, we did modify Mao et al.'s training procedure and architecture in the following ways. First, we used L2-regularization on the LSTM weights. We found the best generalization with an L2 fraction of 10^{-4}. We disregarded dropout as we saw little to no improvement at cost of training time, and we increased the batch size to 32. We also used the Adam Optimizer to train, where Mao et al. use vanilla stochastic gradient descent. This choice also leads to a different learning schedule. We started training with a learning rate of 10^{-3} and dropped the learning rate by an order of magnitude every 20 epochs. We were not able to reproduce Mao et al.'s reported performance of 0.654. Our best performing network achieved a comprehension precision of 0.504.

6.2.1.1 RGB Models

For direct comparison between the SUN-SPOT and Google RefExp, we omit SUN-SPOT's depth channel so that the models can accept both SUN-SPOT and RefExp examples as input. We use a pretrained VGG-16 network [134] to produce image features.

We trained two RGB referring expression networks. The first model, **Baseline**, was trained for 60 epochs on the Google RefExp dataset. For the second model, **Baseline+fine**, we fine-tuned the Baseline model with the SUN-SPOT training set for a further 30 epochs.

6.2.1.2 RGB-D Models

To test the potential gains from adding depth based features, we train a custom VGG-16 network with a 4th channel added to the first convolutional layer of a conventional VGG-16 network.

Because no other referring expression dataset contains RGB-D images, we generated synthetic depth, using MegaDepth [135], for the COCO dataset [2], the source of images for RefExp. To fine-tune the VGG-16 network, we train it for 65 epochs on the portion of the COCO 2014 training set disjoint from the RefExp dataset. During fine-tuning, we use multi-label binary cross-entropy

loss to predict all the object labels in each image.

We train two referring expression models with depth. The first model, **VGG+D**, is trained for 30 epochs on the RefExp training set with the added depth channel. For the second model, **VGG+D+fine**, we fine-tune the VGG+D model on SUN-SPOT for a further 5 epochs. For a direct comparison, we also train an RGB VGG-16 in the same way as the depth networks, **VGG** and **VGG+fine**.

We also experimented with HHA depth preprocessing [29] which is the standard approach for incorporating depth into image networks. However, we observed that HHA depth preprocessing magnified errors in surface normal prediction in synthetic depth images. Additionally, many of the COCO images do not have a ground-plane, which is required to calculate HHA. Therefore we did not find HHA suitable for synthetic depth. See Chapter 2 for more details about depth fusion approaches.

6.2.2 Referring Expression Generation

There has been some debate about the best automatic metrics for measuring the quality of generated expressions. Mao et al. use the ability of the network to localize the generated expression, but this might instead measure the confidence of the network in its own output, and in fact Mao et al. report higher scores for generated expressions than for ground truth [4].

Therefore, we evaluate our generated expressions with automated metrics, BLEU [110], ROUGE-L[5], and CIDEr [111]. Traditionally used for measuring the quality of machine translation and image captioning, they can also be used for comparing the similarity of two referring expressions. We use BLEU1 scores which measure the number of co-occurring unigrams in a pair of sentences. Therefore BLUE1 scores are high when precision is high. As a complementary metric, we also use ROUGE-L which measures the longest common subsequence, and therefore is high when the recall is high. Our final metric, CIDEr, is designed to measure consensus across a set of sentences. In short, it is similar to BLEU in that it measures co-occurring n-grams, but it also uses tf-idf to weight the n-grams so that less common shared n-grams are weighted higher than frequent n-grams.

Model	Dataset	B1	R-L	C	P@1
-	Google RefExp	0.33	0.31	0.82	
-	SUN-SPOT	0.56	0.51	1.33	
Baseline	Google RefExp	0.30	0.31	0.31	0.50
Baseline	SUN-SPOT	0.27	0.19	0.07	0.20
Baseline+fine	Google RefExp	0.18	0.21	0.08	0.50
Baseline+fine	SUN-SPOT	0.45	0.44	0.15	0.33

Table 6.2: Quantitative results for generation and comprehension on RGB models. Columns are BLEU1 (B1), ROUGE-L(R-L), CIDEr(C), and Precision at 1(P@1). The first two rows compare ground truth referring expressions to establish an upper-bound. The other rows evaluate generated sentences.

CIDEr was used as part of the evaluation procedure on the MSCOCO challenge [111]. We use implementations of these metrics provided with [136].

These metrics, BLEU and ROUGE in particular, have been criticized for not recognizing semantically equivalent sentences. However, ranking sentences using BLEU is shown to correlate well with human judgment [110], so while the sentence level ratings may vary in accuracy, a corpus level score can show trends in referring expression generation quality.

Table 6.2 shows a summary of the automated similarity metrics. The first two rows show the datasets' internal similarity across referring expressions describing the same object. To calculate this value, we held out one expression from each set of expressions describing the same object. These scores can be considered upper bounds on what generated expressions can achieve as they represent the variance between human annotators. They also show that SUN-SPOT has more internal similarity than Google RefExp by all three metrics.

For all generated expressions, the scores are highest when the expressions are generated for the same dataset that the model was trained on. This is to be expected as the models will learn subtle dataset bias such as the preference for certain vocabulary. This tells us that Google RefExp and SUN-SPOT do have significant biases that stymie transfer learning from one to the other for generating expressions. These biases could stem from different vocabulary or from different sentence structure. It is nevertheless impressive that the Baseline+Fine shows such improved performance on SUN-SPOT despite the relatively small size of that dataset.

6.2.3 Referring Expression Comprehension

The models generate referring expressions, but we can also use them to measure the comprehension of referring expressions by ranking the likelihood of generating the input expression. Generation likelihood ranking was introduced simultaneously by Mao et al. [4] and Hu et al. [108] and has been widely used since to use generative networks for comprehension [104, 99]. To compute generation likelihood, for each target RE, S, we select the bounding box, R^*, which maximizes the probability of generating the target expression for the given image I. This can be expressed as

$$R^* = \arg\max_{R \in C} p(R|S, I) \tag{6.1}$$

where C is the set of all bounding boxes in image I.

In a fully automated scenario, the bounding boxes would be generated by a bounding box proposal system. To estimate an upper-bound on the performance, we use the ground truth bounding boxes. As a metric, we use comprehension precision at 1(P@1), which measures whether the correct bounding box has the highest generation likelihood for a given expression. We compare 8 ground truth bounding boxes per image in the Google RefExp dataset and 10 bounding boxes per image in the SUN-SPOT dataset. Randomly selecting a bounding box would yield 12% precision@1 for the Google RefExp dataset and 10% for the SUN-SPOT dataset.

We report the precision for the comprehension task in Table 6.2. The precision for the Google RefExp dataset does not drop after fine-tuning. This suggests that the SUN-SPOT fine-tuning leads to better generalization for the comprehension task.

6.2.4 Effects of Depth

Table 6.3 compares RGB and RGB-D models. The results for VGG and VGG+D are similar. The addition of a synthetic depth channel has a limited effect on performance. However, between the fine-tuned models, we see a significant improvement in VGG+D+fine, trained with real depth measurements. Real depth measurements, from SUN-SPOT, improve both generation accuracy and comprehension precision. This is an interesting result as it underscores the value of RGB-D datasets

Model	Dataset	B1	R-L	C	P@1
VGG	Google RefExp	0.17	0.19	0.15	0.25
VGG+D	Google RefExp	0.18	0.20	0.21	0.25
VGG+fine	SUN-SPOT	0.30	0.35	0.11	0.13
VGG+D+fine	SUN-SPOT	0.34	0.34	0.14	0.17

Table 6.3: Results comparing RGB image features (VGG and VGG+fine) to RGB-D image features (VGG+D and VGG+D+fine). Metrics are the same as used in Table 6.2.

in building multi-model referring expression models.

6.3 Conclusion

There is a growing demand for interpretable referring expressions networks in human robot interaction. While there has been some initial work in learning relationships between target and landmark objects using attention modules [14, 120], more can be done in this direction. SUN-SPOT can be used to build models that identify both target and landmark objects in the scene and classify the relationship between them. Our open-source baseline networks show that SUN-SPOT may help networks generalize in the referring expression comprehension task. Future work includes leveraging the depth channel of SUN-SPOT, expanding annotation to support more applications, and training interpretable models of spatial relationships. One capability uniquely promoted by our dataset is the development of solutions for online learning of objects in cluttered environments through the proper parsing of spatial referring expressions or discriminative question-answering supported by visual data. Finally, SUN-SPOT will support more rigorous involvement of spatial referring expressions in visual question-answering leading to improved performance.

Chapter 7

Conclusion

This work has discussed two different applications of multimodal learning; RGB-D fusion and referring expression understanding. For the first, I showed that RGB-D features may be most valuable in environments where visual features are degraded. For the second, I introduced a new dataset which focuses on spatial relationships between objects. This dataset, SUN-SPOT, also is the first referring expressions dataset with RGB-D images, opening the door to modeling the depth dependencies of spatial relationships.

The involvement of RGB-D in both projects has produced some complementary results. Both projects found that an early-fusion 4 channel RGB-D feature representation was the most effective, compared to HHA or Depth Aware Convolutional features, despite differing downstream tasks. This is in line with a growing body of work which suggests that HHA representations are not an effective transfer learning technique [31] and that transfer learning has diminishing gains for large datasets and long learning schedules [137].

At the same time, both projects have shown that it is challenging to train models to use depth to produce better results than RGB imagery alone. In part, this may be because RGB is already very feature rich. As shown by monocular depth prediction networks, depth is so correlated with RGB that the later can be used to predict the former. Additionally, the pretraining schemes used by state-of-the-art RGB networks are so sophisticated that replicating the parameter tuning, multi-GPU training, and vast datasets is unlikely for RGB-D without the resources of a technology giant. While my results have not outperformed any benchmarks, I have shown that with the same

training resources depth can improve performance.

I attribute some of the success of the RGB-D fusion work to the shared matrix representation of dense depth maps and RGB images. This commonality made it easier to modify existing CNN architectures and stand on the shoulders of semantic segmentation and object detection researchers who have been chasing and crushing benchmarks for many years. On the referring expressions side of things, how to align linguistic and visual semantic representations is still an open research question. Ongoing work with SUN-SPOT investigates graph-based multimodal representations of spatial referring expressions using abstract meaning representations[138], scene graphs[139], and graph convolutional networks.

Tools like graph convolutional networks are opening up deep learning to new types of structured data, providing even more opportunities to learn from heterogeneous data collections. In our increasingly data rich operation environments, multimodal learning is ripe for new developments. I believe that multimodal learning will be a key ingredient in general purpose artificial intelligence, enabling systems to communicate and reason abstractly about different perceptive modalities.

Bibliography

[1] A. Gatt, I. Van Der Sluis, and K. Van Deemter, "Evaluating algorithms for the Generation of Referring Expressions using a balanced corpus," in Proceedings of the Eleventh European Workshop on Natural Language Generation, 2007.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in European Conference on Computer Vision. Springer, 2014, pp. 740–755.

[3] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg, "ReferItGame: Referring to Objects in Photographs of Natural Scenes," in Conference Empirical Methods in Natural Language Processing, 2014, pp. 787–798.

[4] J. Mao, H. Jonathan, A. Toshev, O. Camburu, A. Yuille, and K. Murphy, "Generation and comprehension of unambiguous object descriptions," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016.

[5] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, S. S. Marie-Francine Moens, Ed. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81.

[6] S. Banerjee and A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments," in Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization. Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2005, pp. 65–72. [Online]. Available: http://www.aclweb.org/anthology/W/W05/W05-0909

[7] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNS," in International Conference on 3D Vision. IEEE, 2017, pp. 11–20.

[8] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in International Conference on Computer Vision, Oct 2017.

[9] "DARPA Subterranean Challenge Artifacts Specification Final Event," Apr. 2021. [Online]. Available: https://subtchallenge.com/resources/SubT_Finals_Artifacts_Specification.pdf

[10] "DARPA Subterranean Challenge Competition Rules Final Event," May 2021. [Online]. Available: https://subtchallenge.com/resources/SubT_Challenge_Finals_Rules.pdf

[11] R. Liu, C. Liu, Y. Bai, and A. L. Yuille, "CLEVR-Ref+: Diagnosing visual reasoning with referring expressions," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4185–4194.

[12] R. Hu, M. Rohrbach, and T. Darrell, "Segmentation from natural language expressions," in European Conference on Computer Vision. Springer, 2016, pp. 108–124.

[13] C. Liu, Z. Lin, X. Shen, J. Yang, X. Lu, and A. Yuille, "Recurrent Multimodal Interaction for Referring Image Segmentation," in International Conference on Computer Vision, 2017.

[14] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko, "Modeling Relationships in Referential Expressions with Compositional Modular Networks," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017.

[15] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 2, pp. 328–341, 2007.

[16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes dataset for semantic urban scene understanding," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016.

[17] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: Structured-light versus time-of-flight kinect," Computer Vision and Image Understanding, vol. 139, pp. 1–20, 2015.

[18] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2015, pp. 567–576.

[19] R. Horaud, M. Hansard, G. Evangelidis, and C. Ménier, "An overview of depth cameras and range scanners based on time-of-flight technologies," Machine Vision and Applications, vol. 27, no. 7, pp. 1005–1020, 2016.

[20] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," IEEE Signal Processing Magazine, vol. 37, no. 4, pp. 50–61, 2020.

[21] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in NIPS, 2014.

[22] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3D scene structure from a single still image," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 824–840, 2008.

[23] F. Khan, S. Salahuddin, and H. Javidnia, "Deep learning-based monocular depth estimation methods—a state-of-the-art review," Sensors, vol. 20, no. 8, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/8/2272

[24] C. Zhao, Q. Sun, C. Zhang, Y. Tang, and F. Qian, "Monocular depth estimation based on deep learning: An overview," Science China Technological Sciences, pp. 1–16, 2020.

[25] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8445–8453.

[26] J. M. U. Vianney, S. Aich, and B. Liu, "RefinedMPL: Refined monocular PseudoLiDAR for 3D object detection in autonomous driving," arXiv:1911.09712, 2019.

[27] F. Fooladgar and S. Kasaei, "A survey on indoor RGB-D semantic segmentation: from hand-crafted features to deep convolutional neural networks," Multimedia Tools and Applications, vol. 79, no. 7, pp. 4499–4524, 2020.

[28] Benchmark Suite – Cityscapes Dataset. [Online]. Available: https://www.cityscapes-dataset.com/benchmarks/

[29] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in European Conference on Computer Vision, 2014, pp. 345–360.

[30] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2013, pp. 564–571.

[31] X. Song, L. Herranz, and S. Jiang, "Depth CNNs for RGB-D scene recognition: Learning from scratch better than transferring from RGB-CNNs," in AAAI Conference on Artificial Intelligence, 2017.

[32] W. Wang and U. Neumann, "Depth-aware CNN for RGB-D segmentation," in European Conference on Computer Vision, 2018, pp. 135–150.

[33] J. Sanchez-Riera, K.-L. Hua, Y.-S. Hsiao, T. Lim, S. C. Hidayati, and W.-H. Cheng, "A comparative study of data fusion for RGB-D based visual recognition," Pattern Recognition Letters, vol. 73, pp. 1–6, 2016.

[34] Z. Gu, L. Niu, H. Zhao, and L. Zhang, "Hard pixel mining for depth privileged semantic segmentation," IEEE Trans on Multimedia, pp. 1–1, 2020.

[35] X. Chen, K.-Y. Lin, J. Wang, W. Wu, C. Qian, H. Li, and G. Zeng, "Bi-directional cross-modality feature propagation with separation-and-aggregation gate for RGB-D semantic segmentation," in European Conference on Computer Vision, 2020, pp. 561–577.

[36] L. Sun, K. Yang, X. Hu, W. Hu, and K. Wang, "Real-time fusion network for RGB-D semantic segmentation incorporating unexpected obstacle detection for road-driving images," IEEE Robotics and Automation Letters, vol. 5, no. 4, pp. 5558–5565, 2020.

[37] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," arXiv:2005.10821, 2020.

[38] K. Sivaraman and A. Murthy, "Object recognition under lighting variations using pre-trained networks," in IEEE Applied Imagery Pattern Recognition Workshop, 2018, pp. 1–7.

[39] Y. P. Loh and C. S. Chan, "Getting to know low-light images with the exclusively dark dataset," Computer Vision and Image Understanding, vol. 178, pp. 30–42, 2019.

[40] L. Neumann, M. Karg, S. Zhang, C. Scharfenberger, E. Piegert, S. Mistr, O. Prokofyeva, R. Thiel, A. Vedaldi, A. Zisserman et al., "NightOwls: A pedestrians at night dataset," in Asian Conference on Computer Vision, 2018, pp. 691–705.

[41] S. Nag, S. Adak, and S. Das, "What's there in the dark," in IEEE International Conference on Image Processing, 2019, pp. 2996–3000.

[42] Y. Xiao, A. Jiang, J. Ye, and M. Wang, "Making of night vision: Object detection under low-illumination," IEEE Access, vol. 8, pp. 123 075–123 086, 2020.

[43] Y. Sasagawa and H. Nagahara, "YOLO in the dark-domain adaptation method for merging multiple models," in European Conference on Computer Vision, 2020.

[44] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 3291–3300.

[45] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in International Conference on Computer Vision, Oct 2017.

[46] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2020.

[47] M. López Antequera. (2020, Aug) Introducing the mapillary planet-scale depth dataset for single-view depth estimation. [Online]. Available: https://blog.mapillary.com/update/2020/08/20/planet-scale-depth-dataset.html

[48] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2016.

[49] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang, "Noise estimation from a single image," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, vol. 1, 2006, pp. 901–908.

[50] J. Lim, J. Kim, J. Sim, and C. Kim, "Robust contrast enhancement of noisy low-light images: Denoising-enhancement-completion," in IEEE International Conference on Image Processing, 2015, pp. 4131–4135.

[51] M. Li, J. Liu, W. Yang, X. Sun, and Z. Guo, "Structure-revealing low-light image enhancement via robust retinex model," IEEE Transactions on Image Processing, vol. 27, no. 6, pp. 2828–2841, 2018.

[52] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "SceneNet RGB-D: Can 5m synthetic images beat generic ImageNet pre-training on indoor segmentation?" in International Conference on Computer Vision, 2017, pp. 2678–2687.

[53] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," arXiv:1706.05587, 2017.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[55] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258.

[56] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2020.

[57] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.

[58] M. T. Ohradzansky, E. R. Rush, D. G. Riley, A. B. Mills, S. Ahmad, S. McGuire, H. Biggie, K. Harlow, M. J. Miles, E. W. Frew, C. Heckman, and S. J. Humbert, "Multi-agent autonomy: Advancements and challenges in subterranean exploration," Journal of Field Robotics, 2021 (Under review).

[59] T. Dang, F. Mascarich, S. Khattak, H. Nguyen, H. Nguyen, S. Hirsh, R. Reinhart, C. Papachristos, and K. Alexis, "Autonomous search for underground mine rescue using aerial robots," in IEEE Aerospace Conference. IEEE, 2020, pp. 1–8.

[60] K. Ebadi, Y. Chang, M. Palieri, A. Stephens, A. Hatteland, E. Heiden, A. Thakur, N. Funabiki, B. Morrell, S. Wood et al., "LAMP: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments," in IEEE International Conference on Robotics and Automation. IEEE, 2020, pp. 80–86.

[61] E. Terry, X. Lei, B. Morrell, S. Daftry, and A.-a. Agha-mohammadi, "Object and gas source detection with robotic platforms in perceptually-degraded environments," in Robotics: Science and Systems Conference Workshop, 2020.

[62] T. Rouček, M. Pecka, P. Čížek, T. Petříček, J. Bayer, V. Šalanskỳ, D. Heřt, M. Petrlík, T. Báča, V. Spurnỳ et al., "DARPA subterranean challenge: Multi-robotic exploration of underground environments," in International Conference on Modelling and Simulation for Autonomous Systems. Springer, 2019, pp. 274–290.

[63] V. Agrawal, "Ground up design of a multi-modal object detection system," Ph.D. dissertation, Carnegie Mellon University Pittsburgh, PA, 2019.

[64] I. D. Miller, F. Cladera, A. Cowley, S. S. Shivakumar, E. S. Lee, L. Jarin-Lipschitz, A. Bhat, N. Rodrigues, A. Zhou, A. Cohen et al., "Mine tunnel exploration using multiple quadrupedal robots," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 2840–2847, 2020.

[65] J. G. Rogers, J. M. Gregory, J. Fink, and E. Stump, "Test your SLAM! the SubT-Tunnel dataset and metric for mapping," in IEEE International Conference on Robotics and Automation. IEEE, 2020, pp. 955–961.

[66] M. Palieri, B. Morrell, A. Thakur, K. Ebadi, J. Nash, A. Chatterjee, C. Kanellakis, L. Carlone, C. Guaragnella, and A.-a. Agha-mohammadi, "LOCUS: A multi-sensor lidar-centric solution for high-precision odometry and 3D mapping in real-time," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 421–428, 2020.

[67] S. S. Shivakumar, N. Rodrigues, A. Zhou, I. D. Miller, V. Kumar, and C. J. Taylor, "PST900: RGB-thermal calibration, dataset and segmentation network," in IEEE International Conference on Robotics and Automation. IEEE, 2020, pp. 9441–9447.

[68] S. Khattak, H. Nguyen, F. Mascarich, T. Dang, and K. Alexis, "Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments," in International Conference on Unmanned Aircraft Systems. IEEE, 2020, pp. 1024–1029.

[69] A. Kramer, M. Kasper, and C. Heckman, "VI-SLAM for subterranean environments," in Conference on Field and Service Robotics. Springer, 2021, pp. 159–172.

[70] G. Jocher, A. Stoken, J. Borovec, NanoCode012, A. Chaurasia, TaoXie, L. Changyu, A. V, Laughing, tkianai, yxNONG, A. Hogan, lorenzomammana, AlexWang1900, J. Hajek, L. Diaconu, Marc, Y. Kwon, oleg, wanghaoyang0106, Y. Defretin, A. Lohia, ml5ah, B. Milanko, B. Fineran, D. Khromov, D. Yiwei, Doug, Durgesh, and F. Ingham. (2021, Apr.) ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations. [Online]. Available: https://doi.org/10.5281/zenodo.4679653

[71] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[72] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of cnn," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 390–391.

[73] A. Z. Taddese, A. Bencomo, A. Maio, A. Schang, A. Larkin, C, C. Aguero, C. Kanellakis, Docker, I. Chen, I. Akbar, J. Li, J. Shepherd, J. Fink, J. L. Rivero, L. Poubel, M. Zhang, M. Huck, M. Dlouhy, M. Pecka, M. Petrlik, M. Carroll, M. X. Grey, N. Koenig, N. Johnson, P. Cizek, S. Zimmermann, S. Loretz, S. Peters, T. Idota, T. Molnar, U. user 616e0, W. Schwab, acschang, bencomo, bfotheri, caguero, chapulina, derekvasquez, hector_escobar, iche033, jonfink, knoedler, md, nkoenig, rlanglois, and robotikacz. (2013) Darpa subt virtual competition software. [Online]. Available: https://github.com/osrf/subt

[74] J. Ku, A. Harakeh, and S. L. Waslander, "In defense of classical image processing: Fast depth completion on the cpu," in Conference on Computer and Robot Vision. IEEE, 2018, pp. 16–22.

[75] J. I. Saeed, Semantics. New York, USA: John Wiley & Sons, Incorporated, 2015. [Online]. Available: http://ebookcentral.proquest.com/lib/ucb/detail.action?docID=4035324

[76] T. Williams and M. Scheutz, "Referring Expression Generation under Uncertainty: Algorithm and Evaluation Framework," in International Natural Language Generation Conference, 2017.

[77] L. Kunze, T. Williams, N. Hawes, and M. Scheutz, "Spatial Referring Expression Generation for HRI : Algorithms and Evaluation Framework," in Artificial Intelligence for Human-Robot Interaction, 2017. [Online]. Available:

https://pdfs.semanticscholar.org/40c1/94346a562e4a94ec57f4f380dd4cd39f5dc7.pdfhttps://www.researchgate.net/profile/Tom{_}Williams19/publication/320084050{_}Spatial{_}Referring{_}Expression{_}Generation{_}for{_}HRI{_}Algorithms{_}and{_}Evaluation{_}Framework/links/59cd13d74585

[78] S. Tellex, R. A. Knepper, A. Li, D. Rus, and N. Roy, "Asking for Help Using Inverse Semantics," in Robotics Science and Systems, 2014. [Online]. Available: http://www.roboticsproceedings.org/rss10/p24.pdf

[79] R. A. Knepper, S. Tellex, A. Li, N. Roy, and D. Rus, "Recovering from failure by asking for help," Autonomous Robots, vol. 39, no. 3, pp. 347–362, 2015.

[80] H. De Vries, F. Strub, S. Chandar, O. P. Deepmind, H. L. Twitter, and A. Courville, "Guess-What?! Visual object discovery through multi-modal dialogue," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017.

[81] Y. Li, C. Huang, X. Tang, and C.-C. Loy, "Learning to Disambiguate by Asking Discriminative Questions," in International Conference on Computer Vision, 2017. [Online]. Available: http://arxiv.org/abs/1708.02760

[82] E. Krause, T. Williams, M. Zillich, and M. Scheutz, "Learning to recognize novel objects in one shot through human-robot interactions in natural language dialogues," in National Conference on Artificial Intelligence, vol. 4, 2014, pp. 2796–2802.

[83] M. R. Walter, S. Hemachandra, B. Homberg, S. Tellex, and S. Teller, "Learning Semantic Maps from Natural Language Descriptions," in Robotics Science and Systems, 2013, pp. 1–8. [Online]. Available: papers3://publication/uuid/77B21DAA-4F22-4E45-96A5-F134DA3457B7

[84] I. Mani and J. Pustejovsky, Interpreting Motion: Grounded Representations for Spatial Language. OUP Oxford, Feb. 2012, google-Books-ID: BpiY8xAEtX8C.

[85] A. Gatt and E. Krahmer, "Survey of the State of the Art in Natural Language Generation : Core tasks , applications and evaluation," Artificial Intelligence Research, vol. 61, pp. 1–111, 2018.

[86] E. Krahmer and K. Van Deemter, "Computational generation of referring expressions: A survey," Computational Linguistics, vol. 38, no. 1, pp. 173–218, 2012.

[87] H. P. Grice, "Logic and conversation," in Syntax and Semantics 3: Speech Arts, P. Cole and J. Morgan, Eds. Academic Press, 1975, pp. 41–58.

[88] R. Dale, "Cooking up referring expressions," in Annual Meeting of the Association for Computational Linguistics. Vancouver, British Columbia, Canada: Association for Computational Linguistics, Jun. 1989, pp. 68–75. [Online]. Available: http://www.aclweb.org/anthology/P89-1009

[89] ——, "Generating Referring Expressions Involving Relations," in Conference on the European Chapter of the Association for Computational Linguistics, 1991.

[90] J. Viethen and R. Dale, "GRE3D7: A Corpus of Distinguishing Descriptions for Objects in Visual Scenes," in Proceedings of the UCNLG+Eval: Language Generation and Evaluation Workshop, 2011, pp. 12–22.

[91] K. Van Deemter, I. Van Der Sluis, and A. Gatt, "Building a semantically transparent corpus for the generation of referring expressions," in International Natural Language Generation Conference, 2006. [Online]. Available: http://www.csd.abdn.ac.uk/

[92] I. Paraboni, M. R. Galindo, and D. Iacovelli, "Stars2: a corpus of object descriptions in a visual domain," Language Resources and Evaluation, vol. 51, no. 2, pp. 439–462, 2017.

[93] A. D. F. Clarke, M. Elsner, and H. Rohde, "Where's Wally: the influence of visual salience on referring expression generation." Frontiers in psychology, vol. 4, no. June, p. 329, 2013.

[94] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick, "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning," in IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2017, pp. 1988–1997.

[95] D. Prendergast and D. Szafir, "Improving object disambiguation from natural language using empirical models," in International Conference on Multimodal Interaction, ser. ICMI '18. New York, NY, USA: ACM, 2018, pp. 477–485. [Online]. Available: http://doi.acm.org/10.1145/3242969.3243025

[96] J. Viethen, "The Use of Spatial Relations in Referring Expression Generation," in International Natural Language Generation Conference, 2008.

[97] H. A. E. Viethen, "The generation of natural descriptions: corpus-based investigations of referring expressions in visual domains," Ph.D. dissertation, Macquarie University, 2011.

[98] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, "Visual7W: Grounded Question Answering in Images," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016.

[99] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, "Modeling context in referring expressions," in European Conference on Computer Vision. Springer, 2016, pp. 69–85.

[100] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei, "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations," International Journal of Computer Vision, vol. 123, pp. 32–73, 2017.

[101] 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," International Journal of Computer Vision, vol. 123, no. 1, pp. 74–93, 2017.

[102] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler, "What are you talking about? Text-to-Image Coreference Chen," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2014. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4278665

[103] C. Gan, Y. Li, H. Li, C. Sun, and B. Gong, "VQS: Linking Segmentations to Questions and Answers for Supervised Attention in VQA and Question-Focused Semantic Segmentation," in International Conference on Computer Vision, 2017.

[104] A. B. Vasudevan, D. Dai, and L. Van Gool, "Object referring in visual scene with spoken language," in IEEE/CVF Winter Conference on Applications of Computer Vision. IEEE, 2018, pp. 1861–1870.

[105] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments," in NIPS 2017 Workshop on Visually-Grounded Interaction and Language, 2017.

[106] L. Yu, H. Tan, M. Bansal, and T. L. Berg, "A Joint Speaker-Listener-Reinforcer Model for Referring Expressions," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017.

[107] J. Liu, L. Wang, and M.-H. Yang, "Referring Expression Generation and Comprehension via Attributes," in International Conference on Computer Vision, 2017, pp. 4866–4874.

[108] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell, "Natural language object retrieval," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 4555–4564.

[109] V. K. Nagaraja, V. I. Morariu, and L. S. Davis, "Modeling context between objects for referring expression understanding," in European Conference on Computer Vision. Springer, 2016, pp. 792–807.

[110] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in Annual Meeting of the Association for Computational Linguistics, 2002.

[111] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in IEEE/CVF Conference on Computer Vision and Pattern Recognition. Boston, MA, USA: IEEE, Jun. 2015, pp. 4566–4575. [Online]. Available: http://ieeexplore.ieee.org/document/7299087/

[112] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-RNN)," in International Conference on Learning Representations, 2015.

[113] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in International Conference on Machine Learning, 2014, pp. 595–603.

[114] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[115] R. Girshick, "Fast R-CNN," in International Conference on Computer Vision, 2015, pp. 1440–1448.

[116] M. Shridhar and D. Hsu, "Grounding spatio-semantic referring expressions for human-robot interaction," in RSS 2017 Workshop on Spatial-Semantic Representations in Robotics, 2017.

[117] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, "Grounding of textual phrases in images by reconstruction," in European Conference on Computer Vision. Springer, 2016, pp. 817–834.

[118] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding," in Conference Empirical Methods in Natural Language Processing, 2016.

[119] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[120] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg, "MAttNet: Modular Attention Network for Referring Expression Comprehension," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016.

[121] C. Mauceri, M. Palmer, and C. Heckman, "SUN-Spot: An RGB-D dataset with spatial referring expressions," in International Conference on Computer Vision Workshop, Oct 2019.

[122] H. J. Escalante, C. A. Hernández, J. A. Gonzalez, A. López-López, M. Montes, E. F. Morales, L. E. Sucar, L. Villasenor, and M. Grubinger, "The segmented and annotated IAPR TC-12 benchmark," Computer Vision and Image Understanding, vol. 114, no. 4, pp. 419–428, 2010.

[123] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," IEEE Transactions on Robotics, vol. 30, no. 2, pp. 289–309, April 2014.

[124] M. Schwarz, A. Milan, A. S. Periyasamy, and S. Behnke, "RGB-D object detection and semantic segmentation for autonomous manipulation in clutter," The International Journal of Robotics Research, vol. 37, no. 4-5, pp. 437–451, 2018. [Online]. Available: https://doi.org/10.1177/0278364917713117

[125] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 1746–1754.

[126] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3D graph neural networks for RGBD semantic segmentation," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5199–5208.

[127] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell, "A category-level 3d object dataset: Putting the kinect to work," in Consumer depth cameras for computer vision. Springer, 2013, pp. 141–165.

[128] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using SFM and object labels," in International Conference on Computer Vision, 2013, pp. 1625–1632.

[129] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in European Conference on Computer Vision, 2012, pp. 746–760.

[130] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask Your Neurons: A Deep Learning Approach to Visual Question Answering," in International Journal of Computer Vision, vol. 125, 2017, pp. 110–135. [Online]. Available: https://arxiv.org/pdf/1605.02697.pdfhttp://arxiv.org/abs/1605.02697

[131] A. Bhattacharyya, M. Palmer, C. Mauceri, and C. Heckman, "Improving cross-modal retrieval with semantic role labels," in International Conference on Computer Vision Workshop, 10 2021 (Submitted), (Submitted).

[132] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. (2020) spaCy: Industrial-strength Natural Language Processing in Python. [Online]. Available: https://doi.org/10.5281/zenodo.1212303

[133] R. Weischedel, M. Palmer, M. Marcus, E. Hovy, S. Pradhan, L. Ramshaw, N. Xue, A. Taylor, J. Kaufman, M. Franchini, M. El-Bachouti, R. Belvin, and A. Houston. (2013) OntoNotes Release 5.0. [Online]. Available: https://hdl.handle.net/11272.1/AB2/MKJJ2R

[134] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[135] Z. Li and N. Snavely, "MegaDepth: Learning single-view depth prediction from internet photos," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[136] S. Sharma, L. E. Asri, H. Schulz, and J. Zumer, "Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation," arXiv preprint arXiv:1706.09799, 2017.

[137] K. He, R. Girshick, and P. Dollar, "Rethinking imagenet pre-training," in International Conference on Computer Vision, October 2019.

[138] J. Bonn, M. Palmer, Z. Cai, and K. Wright-Bettner, "Spatial AMR: Expanded spatial annotation in the context of a grounded Minecraft corpus," in Language Resources and Evaluation Conference. Marseille, France: European Language Resources Association, May 2020, pp. 4883–4892. [Online]. Available: https://www.aclweb.org/anthology/2020.lrec-1.601

[139] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased scene graph generation from biased training," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3716–3725.

[140] T. M. Howard, S. Tellex, and N. Roy, "A natural language planner interface for mobile manipulators," in IEEE International Conference on Robotics and Automation. IEEE, 2014, pp. 6652–6659.

Appendix A

Glossary

A.1 Glossary

collective reference A referring expression indicating a group of objects.

constant reference Referring expressions which only have one possible identity. Usually proper nouns, for example "New York City" [75].

distractor Objects with similar characteristics as the referent that need to be discriminated from the referent by the referring expression [86].

extension All objects which could complete a variable referring expression [75].

grounding or visual grounding The linguistic definition of grounding is assigning meaning to symbols [75]. **Visual grounding** is often used as a synonym for referring expression comprehension. However, **visual grounding** can also be used to describe finding image context for a broader range of natural language including question answering [103]. [140] defines it as assigning physical meaning or context to natural language..

landmark or ground An object mentioned in the referring expression that is not the referent. Usually, the landmark has a relationship to the referent. For example, the landmark object

can be used to anchor a spatial relationship (i.e. to the left of), or a comparative relationship (i.e. bigger than). . The term **ground** comes from Gestalt Psychology. **Landmark** is among more frequent use in the NLP and Computer Vision communities..

referent or target or figure The object indicated by a referring expression. **Referent** the preferred term in NLP although **target** is also used [75]. **Target** shows up more frequently in Computer Vision, probably via image retrieval. **Figure** comes from Gestalt Psychology [84].

referring expression A noun phrase which uniquely identifies an object or set of objects.

referring expression comprehension or localization Given a referring expression, identify the referent. This usually involves selecting a region from an image, either a coordinate, bounding box, or segmentation..

referring expression generation Given a desired referent and the scene context, produce a referring expression..

text-to-image coreference A task where every entity mentioned in an image caption is linked to an image region [101, 102]. **Text-to-image coreference** is referring expression comprehension in the domain of image captions.

variable reference Referring expressions which depend on context. For example, "the large book" identifies a unique object, but in a different scene, the same expression may be used to refer to a different book [75].

Appendix B

Code Repositories

This appendix contains links to and abstracts of the code repositories that support the work described in this thesis. Some of these repositories are completely my work, while others are open-source implementations of research code that I found useful. The majority are forks of research code that I adapted to my own applications. I do my best to attribute authorship accordingly.

B.1 Data Management Tools

B.1.1 SUNRGBD to COCO annotation format

Author: Cecilia Mauceri

URL: `https://github.com/crmauceri/SUNRGBD_COCO`

The SUNRGBD dataset [18] is distributed as a large directory structure including RGB images and depth images. Also included in the distribution is the SUNRGBD toolbox, a set of Matlab scripts and three MAT files with the annotations including semantic segmentation, 2d bounding boxes, and 3d bounding boxes.

This repository contains a set of scripts to convert these SUNRGBD Matlab files to COCO annotation format. The COCO dataset [2] defines a json-structured data format which can represent several different annotation formats. Many dataset loaders exist for coco such as the "cocoapi" which make it a convenient annotation format to use.

B.1.2 RGB-D Data Loaders

Author of RGB Data-Loader: Jianfeng Zhang

Adapted to RGB-D by Cecilia Mauceri

URL: `https://github.com/crmauceri/dataset_loaders`

This repository contains a pytorch dataloader which is capable of loading the RGB-D datasets, Cityscapes, SUNRGBD, and SceneNetRGBD.

B.1.3 SUN-Spot Data Loader

Author of Referring Expressions Data Loader: Licheng Yu

Adapted to SUN-Spot by Cecilia Mauceri

URL: `https://github.com/crmauceri/refer_sunspot`

This repository contains a pytorch dataloader which is capable of loading the SUN-Spot dataset.

B.2 Data Annotation Tools

B.2.1 SUN-Spot Image/Caption Server

Author: Cecilia Mauceri

URL: `https://github.com/crmauceri/caption_api`

The annotation of images with referring expressions for SUN-Spot was achieved using the survey website Qualtrics (`www.qualtrics.com`). This repository contains the API used to populate the survey with images (a Flask app) as well as notes on how to integrate the API with the survey. I hope this will prove useful to researchers hoping to build their own simple annotation pipelines.

B.3 Data Processing

B.3.1 Lidar to Image Plane Projection

Author: Cecilia Mauceri

URL: `https://github.com/crmauceri/lidar2depth`

A ROS node which projects the lidar 3D point cloud onto the camera's image plane to create a sparse depth map.

B.3.2 Monocular Depth Prediction

Author: Zhengqi Li

URL: `https://github.com/zhengqili/MegaDepth`

I used this repository to generate depth for the COCO dataset as described in Chapter 6.

B.3.3 Lidar Completion

Author: Jason Ku

ROS Node: Cecilia Mauceri

URL: `https://github.com/crmauceri/ip_basic`

I used this repository to generate depth completed lidar as described in Chapter 4.

B.4 Networks

B.4.1 Depth Aware CNN

Author: Cecilia Mauceri

URL: `https://github.com/crmauceri/DepthAwareCNN-pytorch1.5`

This repository is a Pytorch 1.5 compatible CUDA implementation of Depth Aware CNN [32]. While an implementation of Depth Aware CNN by the authors of the paper exists, it was broken when the method of building Pytorch C++ extensions was updated in Pytorch 1.0.

B.4.2 RGB-D Semantic Segmentation

Author of Pytorch Deeplab Implementation: Jianfeng Zhang

Adapted for RGB-D by Cecilia Mauceri

URL: `https://github.com/crmauceri/rgbd_deeplab`

This is the code used for Chapter 3.2.

B.4.3 RGB-D Object Detection

YoloV5 implementation by Ultralytics

Adapted for RGB-D by Cecilia Mauceri

`https://github.com/crmauceri/rgbd_yolo`

This is the code used for Chapter 4

B.4.4 Referring Expression Generation and Comprehension

Author: Cecilia Mauceri

`https://github.com/crmauceri/ReferringExpressions`

This is the code used for Chapter 6.2

B.4.5 CNN Visualization

Author: Utku Ozbulak

`https://github.com/utkuozbulak/pytorch-cnn-visualizations`

This is a very useful collection of CNN visualization functions. I used it for the activation visualizations in Chapter 3.3.