

October 2021

Understanding of Visual Domains via the Lens of Natural Language

Chenyun Wu
University of Massachusetts Amherst

Follow this and additional works at: https://scholarworks.umass.edu/dissertations_2



Part of the [Artificial Intelligence and Robotics Commons](#)

Recommended Citation

Wu, Chenyun, "Understanding of Visual Domains via the Lens of Natural Language" (2021). *Doctoral Dissertations*. 2385.

<https://doi.org/10.7275/24569203> https://scholarworks.umass.edu/dissertations_2/2385

This Open Access Dissertation is brought to you for free and open access by the Dissertations and Theses at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

Understanding of Visual Domains via the Lens of Natural Language

Chenyun Wu

Follow this and additional works at: https://scholarworks.umass.edu/dissertations_2



Part of the [Artificial Intelligence and Robotics Commons](#)

**UNDERSTANDING OF VISUAL DOMAINS VIA THE LENS OF
NATURAL LANGUAGE**

A Dissertation Outline Presented

by

Chenyun Wu

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2021

College of Information and Computer Science

© Copyright by Chenyun Wu 2021

All Rights Reserved

UNDERSTANDING OF VISUAL DOMAINS VIA THE LENS OF NATURAL LANGUAGE

A Dissertation Outline Presented

by

Chenyun Wu

Approved as to style and content by:

Subhransu Maji, Chair

Erik Learned-Miller, Member

Mohit Iyyer, Member

Zhe Lin, Member

James Allan, Chair of the Faculty
College of Information and Computer Science

ACKNOWLEDGMENTS

The journey of Ph.D. is exciting and challenging with ups and downs, and it's my great fortune to have my advisor, Professor Subhransu Maji, to support and guide me throughout the whole journey. His talents, enthusiasm, and knowledge in research encouraged me to dive into the field of computer vision and guided me to explore my own path of research. He has always been professional and considerate to provide all the help and guidance I need in different dimensions. I'd also like to thank Professor Erik Learned-Miller, especially for his inspiring ideas and discussions in collaborations and group meetings.

I'm grateful for my internship mentors, Xiaohui Shen, Xiaojie Jin, and Longyin Wen from ByteDance, Zhe Lin and Scott Cohen from Adobe, and Nick Johnston, George Toderici, David Minnen, and Michele Covell from Google. They showed me the scope of research in the industry which plays a key role in shaping my career goal. I'd also like to thank my collaborators, Jong-Chyi Su, Huaizu Jiang, Mikayla Timm, Tsung-Yu Lin, Joydeep Biswas, from whom I learned a lot and gained tons of support.

I'm very lucky to meet our lab mates, Zezhou Cheng, Hang Su, Zitian Chen, Haibin Huang, Yang Zhou, Zhan Xu, Difan Liu, Zhaoliang Lun, Souyoung Jin, Aruni RoyChowdhury, Pia Bideau, Li Yang Ku, Ashish Singh, Matheus Gadelha, Gopal Sharma, Gustavo Pérez. I'd like to thank them for their help and accompany in both research and everyday life. I'm also super thankful for my friends in Amherst: Xiang Li, Chenghao Lyu, Han Li, Zhiqi Huang, Pengshan Cai, Dongxu Zhang, Tongyi Cao, Qingyao Ai, Yue Wang, Dan Zhang, Keping Bi, Xiaoyi Wu, Yuxin Liu, Zhuojun Duan, Fuqian Sun, Yirou Luo, Shuaimin Kang, Jun Wang, Li Wang, and a lot of others. They made my six years in Amherst full of happiness and great memories and provided the best support especially

during my first year in Amherst when I was away from my country the first time, and during the hardest times in the Covid-19 pandemic.

Finally, I have my most sincere gratefulness for my dearest parents. They always support me to pursue my goal with no hesitation, believe in my capability and potential without any doubt, fully accept my failure and are always there to provide comfort and encouragement. It's their unconditioned love that gives me the strength and courage to keep moving forward.

ABSTRACT

UNDERSTANDING OF VISUAL DOMAINS VIA THE LENS OF NATURAL LANGUAGE

SEPTEMBER 2021

Chenyun Wu

B.Sc., PEKING UNIVERSITY

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Subhansu Maji

A joint understanding of vision and language can enable intelligent systems to perceive, act, and communicate with humans for a wide range of applications. For example, they can assist a human to navigate in an environment, edit the content of an image through natural language commands, or search through image collections using natural language queries. In this thesis, we aim to improve our understanding of visual domains through the lens of natural language. We specifically look into (1) images of categories within a fine-grained taxonomy such as species of birds or variants of aircraft, (2) images of textures that describe local color, shape, and patterns, and (3) regions in images that correspond to objects, materials, and textures.

In one line of work, we investigate ways to discover a domain-specific language by asking annotators to describe visual differences between instances within a fine-grained taxonomy. We show that a system trained to describe these differences leads to an accurate and interpretable basis for categorization. In another line of work, we investigate the effectiveness of language and vision models for describing textures, a problem that, despite the

ubiquity of textures, has not been sufficiently studied in the literature. Textures are diverse, yet their local nature allows for the description of appearance of a wide range of visual categories. The locality also allows us to systematically generate synthetic variations to investigate how disentangled visual representations are for properties such as shape, color, and figure-ground segmentation. Finally, instead of modeling an image as a whole, we design a system that allows descriptions of regions within an image. A challenge is to handle the long-tail distribution of names and appearances of concepts within natural scenes. We design a modular framework that integrates object detection, semantic segmentation, and contextual reasoning with language that leads to better performance. In addition to methods and analysis, we contribute datasets and benchmarks to evaluate the performance of models in each of these domains.

The availability of large-scale pre-trained models for vision (e.g., ResNet [47]) and language (e.g., BERT [35]) have catalyzed improvements and novel applications in computer vision and natural language processing, but until recently similar models that could jointly reason about language and vision were not available. This has changed through the availability of models such as CLIP [94], which have been trained on a massive number of images with associated texts. Therefore, we analyze the effectiveness of CLIP-based representations for tasks posed in our earlier work. By comparing and contrasting these with domain-specific ones we presented in the earlier chapters, we shed some light on the nature of the learned representations and the biases they encode.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	iv
ABSTRACT	vi
LIST OF TABLES	xi
LIST OF FIGURES	xiv
 CHAPTER	
1. INTRODUCTION AND BACKGROUND	1
1.1 Attribute-based Representations in Computer Vision	4
1.2 Datasets for Vision and Language	6
1.2.1 Visual referring datasets	6
1.3 Methods for Vision and Language	8
1.3.1 Approaches for grounding language to image regions	9
1.3.2 Large-scale pre-training of image and text representations	10
2. DESCRIBING FINE-GRAINED CATEGORIES	12
2.1 A Dataset of Attribute Phrases	15
2.2 Models	16
2.2.1 Speaker models	16
2.2.2 Listener models	18
2.3 Results	19
2.3.1 Evaluating listeners	19
2.3.2 Evaluating speakers	21
2.3.3 Fine-grained classification with attributes	23

2.3.4	Visualizing the space of descriptive attributes	25
2.3.5	Image retrieval with descriptive attributes.....	28
2.3.6	Generating attribute explanations.....	28
2.4	Summary	29
3.	DESCRIBING TEXTURES	30
3.1	Dataset and Tasks	32
3.1.1	Dataset collection	33
3.1.2	Tasks and evaluation metrics	34
3.2	Methods	36
3.2.1	A discriminative classification approach	36
3.2.2	A metric learning approach	37
3.2.3	A generative language approach	39
3.3	Experiments and Analysis	40
3.3.1	Phrase and image retrieval	40
3.3.2	Description generation	41
3.3.3	A critical analysis of language modeling	42
3.4	Applications.....	46
3.4.1	Describing textures of fine-grained categories	46
3.4.2	Fine-grained classification with texture attributes	47
3.5	Summary	48
4.	DESCRIBING REGIONS IN IMAGES	49
4.1	The VGPHRASECUT Dataset	51
4.1.1	Data collection pipeline	52
4.1.2	Dataset statistics.....	54
4.1.3	Evaluation metrics	55
4.2	A Modular Approach to PhraseCut	55
4.3	Results and Analysis	62
4.3.1	Comparison to baselines	62
4.3.2	Ablation studies and analysis	63
4.3.3	Modular heatmap visualization	65
4.3.4	Failure case analysis	65

4.4	Summary	69
5.	EVALUATING LARGE-SCALE LANGUAGE-VISION MODELS	70
5.1	Models and Datasets	71
5.1.1	Overview of CLIP	72
5.1.2	Baselines	73
5.2	Experiments and Analysis	73
5.2.1	Recognizing fine-grained differences between aircraft images	73
5.2.2	Phrase and Image Retrieval on DTD ²	75
5.2.3	Phrase and Image retrieval of texture phrases on CUB Dataset	78
5.2.4	Understanding compositional phrases on synthetic texture images	81
5.2.5	Zero-Shot Classification with Attribute Phrases	83
5.3	Summary	85
6.	CONCLUSION	86
	BIBLIOGRAPHY	87

LIST OF TABLES

Table	Page
<p>1.1 Comparison of visual referring datasets. The proposed VGPHRASECUT dataset has a significantly higher number of categories than RefCOCO and Google RefExp, while also containing multiple instances.</p>	7
<p>2.1 Accuracy (%) of various listeners in the RG using attribute phrases provided by a human speaker.</p>	20
<p>2.2 Accuracy (%) of the simple listener (SL) on RG using human-generated attribute phrases at positions one through five across the validation and test set. The accuracy decreases monotonically from one to five suggesting that the top attribute phrases are easier to discriminate.</p>	20
<p>2.3 Accuracy in the RG using different speakers and listeners. Test represents the full test set consisting of 2350 image pairs. Test* represents a subset of 100 test set image pairs for which we collected human listener results. For the human listener, we report the accuracy when there is a majority agreement, and accuracy with guessing (in brackets). DS is significantly better at generating discriminative attribute phrases than SS.</p>	23
<p>2.4 Accuracy of pragmatic speakers with human listeners on the Test* set. After generating the descriptions by the speaker model (either SS or DS), we use the listener model (SL_r or SL) to rerank them. We report the accuracy based on human listener from the user study. We report both the accuracy when there is majority agreement, and accuracy with guessing (in brackets). Pragmatic speakers are strictly better than non-pragmatic ones.</p>	24
<p>3.1 Performance on phrase retrieval and image retrieval on DTD². “Classifier: Feat x” stands for the classifier with image features from ResNet layer block x (or a concatenation of two layers.) All triplet models in this table are trained with phrase input. Among the language models BERT works the best.</p>	40

3.2	Retrieving textures from descriptions.	40
3.3	Description generation on textures. Synthesizing descriptions from phrases retrieved by the metric-learning based approach outperforms other baselines.	40
3.4	Image retrieval performance of R-Precision on synthetic tasks.	42
4.1	Comparison of various approaches on the entire test set of VGPHRASECUT. We compare different combinations of modules in our approach (HULANet) against baseline approaches: Mask-RCNN, RMI and MattNet.	60
4.2	The mean-IoU on VGPHRASECUT test set for various category subsets. The column <i>coco</i> refers to the subset of data corresponding to the 80 coco categories, while the remaining columns show the performance on the top 100, 101-500 and 500+ categories in the dataset sorted by frequency.	60
4.3	The mean-IoU on VGPHRASECUT test set for additional subsets. <i>att/rel</i> : the subset with attributes/relationship annotations; <i>att+/rel+</i> : the subset which requires attributes or relationships to distinguish the target from other instances of the same category; <i>single/multi/many</i> : subsets that contain different number of instances referred by a phrase; <i>small/mid/large</i> : subsets with different sizes of the target region.	61
5.1	Comparison of model and training data size. Note that the image encoders of OID-SL and DTD2-ML are pre-trained on ImageNet with 14M images. The text encoder of DTD2-ML contains BERT with 110M parameters which is not updated during training, and we only train a linear layer on top of it.	74
5.2	Referring accuracy of CLIP on OID Attribute Phrases validation set with different input text templates. According to the text templates we have evaluated, the templates have a small impact of 1.7% on referring accuracy. “An image of an aircraft with [Phrase]” is the best template we have found.	75
5.3	Image and phrase retrieval mean average precision of CLIP on DTD² validation split. Considering both image and phrase retrieval performance, we select “An image of [Phrase] texture” as our template for further experiments.	77
5.4	Compare the phrase retrieval and image retrieval performance of DTD2-ML and CLIP on DTD² test set.	81

5.5 **Compare the performance of phrase retrieval and image retrieval with DTD2-ML and CLIP on CUB test set.** We experiment with 17 attributes that are included in both CUB and DTD² as input queries. 81

5.6 **Compare the R-Precision of image retrieval on texture compositional tasks with DTD2-ML and CLIP.** 83

LIST OF FIGURES

Figure	Page
2.1 Reference games with attribute phrases. <i>Left:</i> Each annotation in our dataset consists of five pairs of attribute phrases. <i>Right:</i> A <i>reference game</i> played between a speaker who describes an attribute of an image within a pair and a listener whose goal is to pick the right one.	13
2.2 The interface used to collect five different <i>attribute phrase</i> pairs adapted from [78]. Amazon Mechanical Turkers were paid \$0.12 for annotating three pairs.	16
2.3 Example output of simple speaker SS and discerning speaker DS. Simple speaker takes the left image in the green box as input, while the discerning speaker takes both images as input. In brackets are the probabilities according to the speaker.	22
2.4 An example output of various speakers. Given the image pair, we use SS and DS to generate descriptions of the top left image. Outputs from SS and DS are listed in the order of probabilities from speaker beam search. Outputs of SS+SL _r and DS+SL _r are reranked by SL _r . Green checks mean human listener picks correct image with certain, while question marks mean human listener is uncertain which image is referred to. The results indicate that DS is better than SS, and reranking using listeners improves the quality of top sentences.	25
2.5 Classification accuracy on FGVC aircraft dataset using the 46 dimensional OID attributes and varying number of attribute phrases.	26
2.6 Visualization of the 500 most frequent descriptions. Each attribute is embedded into a 1024 dimensional space using the simple listener SL and projected into two dimensions using t-SNE [110].	26
2.7 Top 18 images ranked by the listener for various attribute phrases as queries (shown on top). We rank the images by the scores from the simple listener on the concatenation of the attribute phrases. The images are ordered from top to bottom, left to right.	27

2.8	Top 10 discriminative attribute phrases for pairs of categories from FGVC aircraft dataset. Descriptions are generated by the discerning speaker for each pair of images in the first and second category. The phrases sorted by the occurrence frequency provides an attribute-based explanation of the visual difference between two categories.	27
3.1	We introduce the Describable Textures in Detail Dataset (DTD²) consisting of texture images from DTD [28] with natural language descriptions, which provide rich and fine-grained supervision for various aspects of texture such as color compositions, shapes, and materials.	31
3.2	Statistics of DTD². The “overall” column in the table shows the statistics of all data, while the “frequent” column only considers the phrases (or words) that occur at least 10 (or 5) times in the training split which forms our evaluation benchmark. The cloud of phrases has the font sizes proportional to square-root of frequencies in the dataset. The vocabulary significantly expands the 47 attributes of DTD.	34
3.3	Retrieve DTD² test images with language input. We show top 5 retrieved images from the classifier, the triplet model with phrase input and with description input. From left to right we show example inputs of (1) phrases the classifier has been trained on, (2) novel phrases beyond the frequent phrase classes, and (3) full descriptions.	41
3.4	Phrase retrieval and description generation on DTD² test images. For each input image, we list ground-truth descriptions beneath, and generated descriptions on the right. For the classifier and the triplet model, we concatenate the top 5 retrieved phrases as the description. Bold words are the ones included in ground-truth descriptions.	42
3.5	Retrieval on synthetic images. Positive images are in dashed blue borders, hard negative ones are in dotted red borders.	43
3.6	Fine-grained categories visualized as their training images (top row), maximal texture images (middle row), and texture attributes (bottom row). The size of each phrase in the cloud is inversely decided by its Euclidean distance to the input maximal texture image calculated by the triplet model.	46

3.7	Classification on CUB dataset with DTD² texture attributes. <i>Left:</i> classification accuracy vs. number of input features. Orange and green markers with the same shape are comparable with the same set of CUB attributes with or without the DTD ² attributes. <i>Right:</i> The phrase clouds display important phrases for a few bird categories. Red phrases correspond to positive weights and blue are negative for a linear classifier for the category. Font sizes represent the absolute value of the coefficient.	47
4.1	PhraseCut task and our approach. PhraseCut is the task of segmenting image regions given a natural language phrase. Each phrase is templated into words corresponding to <i>categories</i> , <i>attributes</i> , and <i>relationships</i> . Our approach combines these cues in a modular manner to estimate the final output.	50
4.2	Example annotations from the VGPHRASECUT dataset. Colors (blue, red, green) of the input phrases correspond to words that indicate attributes, categories, and relationships respectively.	50
4.3	Illustrations of our VGPHRASECUT dataset collection pipeline. Step 1: blue boxes are the sampling result; red boxes are ignored. Step 2: Phrase generation example in the previous image. Step 3: User interface for collecting region masks. Step 4: Example annotations from trusted and excluded annotators. Step 5: Instance label refinement examples. Blue boxes are final instance boxes, and red boxes are corresponding ones from Visual Genome annotations.	52
4.4	Statistics of the VGPHRASECUT dataset. <i>Top row:</i> Word clouds of categories (left), attributes (center), and relationship descriptions (right) in the dataset. The size of each phrase is proportional to the square root of its frequency in the dataset. <i>Bottom row:</i> breakdowns of the dataset into different subsets including contents in phrases (first), category frequency (second), size of target region relative to the image size (third), number of target instances per query phrase (fourth), and types of category (last). The leftmost bar chart shows the breakdown of phrases into those that have category annotation (cat) and those that can be distinguished by category information alone (cat+), and similarly for attributes and relationships.	54

4.5	Architecture of HULANet. The architecture consists of modules to obtain attribute, category, and relation predictions given a phrase and an image. The attribute and category scores are obtained from Mask-RCNN detections and projected back to the image. The scores across categories and attributes are combined using a module-specific attention model. The relationship module is a convolutional network that takes as input the prediction mask of the related category and outputs a spatial mask given the relationship predicate. The modules are activated based on their presence in the query phrase and combined using an attention mechanism guided by the phrase.	56
4.6	Prediction results on VGPHRASECUT dataset. Rows from top to down are: (1) input image; (2) ground-truth segmentation and instance boxes; (3) MattNet baseline; (4) RMI baseline; (5) HULANet (cat + att + rel).	64
4.7	HULANet prediction results and heatmaps on phrases with attributes. Rows from top to down are: (1) input image; (2) ground-truth segmentation and instance boxes; (3) predicted binary mask from HULANet (cat+att+rel); (4) heatmap prediction from the category module; (5) heatmap prediction from the attribute module.	66
4.8	HULANet prediction results and heatmaps on phrases with relationships. Rows from top to down are: (1) input image; (2) ground-truth segmentation and instance boxes; (3) predicted binary mask from HULANet (cat+att+rel); (4) heatmap prediction from the category module; (5) heatmap prediction from the relationship module; (6) heatmap prediction of the supporting object (in the relationship description) from the category module.	67
4.9	Negative results from HULANet on VGPHRASECUT test set. Rows from top to down are: (1) input image; (2) ground-truth segmentation and instance boxes; (3) predicted binary mask from HULANet (cat+att+rel); (4) heatmap prediction from the category module; (5-6) heatmap predictions from additional (attribute or relationship) modules.	68
5.1	Summary of CLIP model. Figure is from [94] Figure 1: “CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset’s classes.”	72

5.2	Phrases with best and worst referring accuracy for CLIP on OID Attribute Phrases referring task. For each phrase P , we calculate the referring accuracy on data when P is one of the two input phrases. Left: 50 phrases with highest accuracy ($\geq 95\%$); Right: 50 phrases with lowest accuracy ($\leq 51\%$). Font size is proportional to the phrase frequency in OID Attribute Phrases Dataset.	76
5.3	Failure cases of CLIP on OID Attribute Phrases referring task. For each example, the ground-truth in the Attribute Phrases Dataset indicates that the first phrase (before “vs/”) describes the image on the left, but CLIP predicts the opposite.	77
5.4	Image retrieval examples on DTD² test set from CLIP and DTD2-ML. For each query attribute phrase, we display 5 random ground-truth images labeled with the given attribute, and the top 5 retrieved images from CLIP and DTD2-ML.	78
5.5	Phrase retrieval examples on DTD² test set from CLIP and DTD2-ML. For each image, we display its ground-truth phrases labeled in DTD ² and top 20 retrieved phrases from CLIP and DTD2-ML. The bolded retrieved phrases are the ones included in the ground truth.	79
5.6	Cloud of phrases with the best and worst performance of image retrieval on DTD² test set for CLIP and DTD2-ML. The blue (red) cloud is sampled from the top 80 phrases with the highest(lowest) average precision. On the right, we also display 80 phrases with the maximum difference in average precision between CLIP and DTD2-ML. Font sizes are proportional to phrase frequencies in DTD ²	80
5.7	Image retrieval examples on CUB test set from CLIP and DTD2-ML. For each query attribute phrase, we display 5 random ground-truth images labeled with the given attribute, and the top 5 retrieved images from CLIP and DTD2-ML.	82
5.8	Image retrieval average precision of each query attribute on CUB test set from CLIP and DTD2-ML. The gray line shows the accuracy difference between CLIP and DTD2-ML. CLIP outperforms DTD2-ML on all attributes except “plain”.	82

5.9 **Examples of constructed category descriptions containing attributes.**

The category names are in bold. Although The three categories have similar names, the attributes can reflect their subtle differences between species, *e.g.*, “**Rhinoceros Auklet**” is more “duck-like” with “buff leg”, “**Parakeet Auklet**” has “white eye” and “white belly”, “**Crested Auklet**” has “crested head” and “black nape”. 83

CHAPTER 1

INTRODUCTION AND BACKGROUND

Vision and language are the fundamental media of perception and communication respectively, thus the joint understanding of the two is essential for an intelligent system to perceive, act, and communicate with humans in various scenarios. For example, a robot can interact and assist a human to navigate in a scene using language, or an automatic system can edit the visual content of images or videos through natural language commands. Thanks to the development of deep neural networks and large-scale datasets, both fields of computer vision and natural language processing have achieved great progress in recent years, making it possible to bring them together and benefit realistic applications.

The goal of this thesis is to achieve a better understanding of visual domains leveraging large-scale and detailed supervision from natural language descriptions. We specifically look into three visual domains: (1) images of categories within a fine-grained taxonomy, (2) images of texture which describes local patterns, (3) objects and stuff regions in natural images. Furthermore, we apply general-domain pre-trained models to specialized domains in a zero-shot manner and compare the performance against smaller models trained on each domain. We demonstrate that by aligning visual representations with language, one can enable several applications such as image retrieval and editing, as well as fine-grained classification with naturally interpretable models.

While the representations vary across domains, we address common benefits and challenges when combining vision and language. Firstly, using natural language makes it possible to collect datasets through crowd-sourcing and require less expertise in specific domains from the annotators. This enables us to collect larger-scale datasets but also results in po-

tentially more noisy annotations. The tasks must be carefully designed such that the joint understanding is required and cannot be circumvented by only one modality or guessing based on statistical bias in datasets. Automatic pre-process and post-process mechanisms also play an important role to guarantee the data quality. Secondly, natural language descriptions can capture detailed attributes of images and objects that are not covered in manually designed set of category labels or binary attributes. Such natural language supervision can improve the modeling of visual details and benefits fine-grained recognition. However, the language vocabulary often follows a long-tail distribution, and it is not mutual exclusive (e.g., an instance can be both a “skier” and a “girl”) with complicated associations between words, which are challenging to model. Lastly, vision and language differ in structure. Vision is high-dimensional with hierarchical semantics from color, texture to objects, and relationships between objects. Language descriptions on different visual semantic levels possess different vocabularies. Language is discrete and compositional (e.g., an entity can be modified by various attributes). It is challenging to align the composition and relationships in language to visual signals.

In Chapter 2, we present a framework for learning to describe fine-grained visual differences between instances using *attribute phrases*. Attribute phrases capture distinguishing aspects of an object (e.g., “propeller on the nose” or “door near the wing” for airplanes) in a compositional manner. Instances within a category can be described by a set of these phrases and collectively they span the space of semantic attributes for a category. We collect a large dataset of such phrases by asking annotators to describe several visual differences between a pair of instances within a category. We then learn to describe and ground these phrases to images in the context of a *reference game* between a speaker and a listener. The goal of a speaker is to describe the attributes of an image that allows the listener to correctly identify it within a pair. We also show that embedding an image into the semantic space of attribute phrases improves fine-grained classification accuracy over

existing attribute-based representations. This work was published as “Reasoning about Fine-grained Attribute Phrases using Reference Games” at ICCV 2017 [106].

For image classification, especially in fine-grained domains, deep neural networks are known to rely largely on recognizing textures. In Chapter 3, we focus on natural language for describing textures, which allows us to exclude the effects of shape, object category, or other high-level cues. Textures in natural images can be characterized by color, pattern, periodicity of elements within them, and other attributes that can be described using natural language. We propose a novel dataset containing rich descriptions of textures and conduct a systematic study of current generative and discriminative models for grounding language to images on this dataset. We find that while these models capture some properties of texture, they fail to capture several compositional properties (*e.g.*, “colors of dots”). Our dataset also allows us to train interpretable models and generate language-based explanations of what discriminative features are learned by deep networks for fine-grained categorization where texture plays a key role. We present visualizations of several fine-grained domains and show that texture attributes learned on our dataset offer improvements over expert-designed attributes. This work was published as “Describing Textures using Natural Language” at ECCV 2020 [121].

In Chapter 4, we extend the referring task to a more realistic setting: instead of selecting one image out of a pair, we consider segmenting image regions given a natural language phrase. Phrases in our proposed dataset correspond to multiple regions and describe a large number of object (*i.e.*, categories such as cars and humans with well defined extent and characteristic shape) and stuff (*i.e.*, categories such as sky and grass with less well defined shape and extent) categories as well as their attributes such as color, shape, parts, and relationships with other entities in the image. Our experiments show that the scale and diversity of concepts in our dataset poses significant challenges to the existing state-of-the-art. We systematically handle the long-tail nature of these concepts and present a modular approach to combine category, attribute, and relationship cues that outperforms existing

approaches. This work was published as “PhraseCut: Language-based Image Segmentation in the Wild” at CVPR 2020 [120].

In Chapters 2, 3 and 4, we have adopted large-scale pre-trained models for vision (e.g., VGG [105] and ResNet [47]) and language (e.g., FastText [20] and BERT [35]) and fine-tune them on specific domains for vision and language tasks. However, this requires data collection and training on each domain, which can be expensive. Recently this has been changed through the availability of models such as CLIP [94] trained on a massive number of images with associated texts across various domains which can serve as a pre-trained benchmark for joint reasoning of vision and language. In Chapter 5, we apply CLIP in a zero-shot manner to fine-grained tasks that have been studied in other chapters, and show that it can reach competitive performance compared against fully supervised models. Benefiting from large-scale training data, it can overcome challenges of understanding language compositionality and domain transfer which are difficult for domain-specific models trained on limited data. It can also perform fine-grained classification where only the category names and attribute are required.

In the remainder of this Chapter, we summarize the related works on attribute representations, as well as vision and language datasets and methods with an emphasis on the referring task.

1.1 Attribute-based Representations in Computer Vision

Attributes have been widely used in computer vision as an intermediate, interpretable representation for high-level recognition. They often represent properties that can be shared across categories, *e.g.*, both a car and a bicycle have wheels, or within a subordinate category, *e.g.*, birds can be described by the shape of their beak. Due to their semantic nature, they have been used for learning interpretable classifiers [41, 40], attribute-based retrieval systems [24], as high-level priors for unseen categories for zero-shot learning [67, 57], and as a means for communication in an interactive recognition system [64].

A different line of work has explored the question of discovering task-specific attributes. Berg *et al.* [18] discover attributes by mining frequent n-grams in captions. Parikh and Grauman [89] ask annotators to name directions that maximally separate the data according to some underlying features. Other approaches [98, 56, 6] have mined phrases from online text repositories to discover commonsense knowledge about the properties of categories (*e.g.*, cars have doors). For a detailed description of the above methods see this recent survey [79].

Attributes have also been used to describe textures. Early works [8, 107, 15] showed that textures can be categorized along a few semantic axes such as “coarseness”, “contrast”, “complexity” and “stochasticity”. Bhusan *et al.* [19] systematically identified words in English that correspond to visual textures and analyzed their relationship to perceptual attributes of textures. This was the basis of the Describable Texture Dataset (DTD) [28] which consolidated a list of 47 texture attributes along with images downloaded from the Internet. The dataset captures attributes such as “dotted”, “chequered”, “honeycombed” and “lined”. However, it does not capture detailed properties, such as the color of the structural elements (“red and green dots”), or the attributes that describe the background color, *etc.*

In Chapter 2, we extend the prior work [78] of collecting attribute phrases in a pairwise manner and propose methods for generating and interpreting attribute phrases. In Chapter 3, we model the rich space of texture attributes in a compositional manner beyond existing binary attributes. We also show that attribute phrases collected from via crowd-sourcing in natural language format describe useful features not included in handicraft binary attributes for fine-grained categorization. In Chapter 5, we use attribute phrases to expand the descriptions of categories beyond their names, and improve the performance of zero-shot fine-grained classification. In Chapter 4, we study the combination of attributes with categories and relationships and the handling of long-tail distribution.

1.2 Datasets for Vision and Language

The vision and language community has put significant efforts into building large-scale datasets. Image captioning datasets such as MS-COCO [70], Flickr30K [127] and Conceptual Captions [103] contain sentences describing the general content of images. The Visual Question Answering dataset [12] provides language question and answer pairs for each image, which requires more detailed understanding of the image content. In visual grounding datasets such as RefClef [60], RefCOCO [81, 129] and Flickr30K Entities [92], detailed descriptions of the target object instances are annotated to distinguish them from other objects.

Besides the datasets collected through crowd-sourcing, there is vast amount of data on the Internet of images paired with tags, captions and descriptions. Recent works [94, 54, 22, 58, 27] leverage internet data through gathering image search results to train large-scale models. In Chapter 5 we provide detailed analysis of [94] in fine-grained domains.

The existing vision-and-language datasets focus on recognizing object categories and descriptions of pose, viewpoint, and their relationships to other objects, and have a limited treatment of attributes related to texture. In Chapter 3, we introduce a novel dataset focused on language descriptions of textures to fill in the gap.

1.2.1 Visual referring datasets

Tasks where annotators are asked to describe an object in an image such that another can correctly identify it provides a way to collect context-sensitive captions [60]. These tasks have been widely studied in the linguistics community in an area called pragmatics (see Grice’s maxims [45]). Our work in Chapter 2 aims to collect and generate referring expressions for fine-grained discrimination between image instances. Referring expression generation has also been extended to interactive dialogue systems [30, 32]. Much prior work in computer vision has focused on generating referring expressions to distinguish an object within an image, as discussed below.

Dataset	ReferIt [60]	Google RefExp [81]	RefCOCO [129]	Flickr30K Entities [93]	Visual Genome [65]	VGPHRASECUT
# images	19,894	26,711	19,994	31,783	108,077	77,262
# instances	96,654	54,822	50,000	275,775	1,366,673	345,486
# categories	-	80	80	44,518	80,138	3103
multi-instance	No	No	No	No	No	Yes
segmentation	Yes	Yes	Yes	No	No	Yes
referring phrase	short phrases	long descriptions	short phrases	entities in captions	region descriptions	templated phrases

Table 1.1: **Comparison of visual referring datasets.** The proposed VGPHRASECUT dataset has a significantly higher number of categories than RefCOCO and Google RefExp, while also containing multiple instances.

Table 1.1 shows a comparison of datasets related to grounding referring expressions to regions in images. The *ReferIt* dataset [60] was collected on images from ImageCLEF using a ReferItGame between two players. Mao *et al.* [81] used the same strategy to collect a significantly larger dataset called *Google RefExp*, on images from the MS COCO dataset [70]. The referring phrases describe objects and refer to boxes inside the image across 80 categories, but the descriptions are long and perhaps redundant. Yu *et al.* [129] instead collect referring expressions using a pragmatic setting where there is limited interaction time between the players to generate and infer the referring object. They collected two versions of the data: *RefCOCO* that allows location descriptions such as “man on the left”, and *RefCOCO+* which forbids location cues forcing a focus on other visual clues. One drawback is that *Google RefExp*, *RefCOCO* and *RefCOCO+* are all collected on *MS-COCO* objects, limiting their referring targets to 80 object categories. Moreover, the target is always one single instance, and there is no treatment of stuff categories.

Another related dataset is the *Flickr30K Entities* [93]. Firstly entities are mined and grouped (co-reference resolution) from captions by linking phrases that describe the same entity and then the corresponding bounding-boxes are collected. Sentence context is often needed to ground the entity phrases to image regions. While there are a large number of categories (44,518), most of them have very few examples (average 6.2 examples per category) with a significant bias towards human-related categories (their top 7 categories are “man”, “woman”, “people”, “shirt”, “girl”, “boy”, “men”). The dataset also does not contain segmentation masks. nor phrases that describe multiple instances.

The *Visual Genome (VG)* dataset [65]. annotates each image as a “scene graph” linking descriptions of individual objects, attributes, and their relationships to other objects in the image. The dataset is diverse, capturing various object and stuff categories, as well as attribute and relationship types. However, most descriptions do not distinguish one object from other objects in the scene, *i.e.*, they are not referring expressions. Also, VG object boxes are very noisy.

In Chapter 4, we introduce our *VGPhraseCut* dataset which pushes the grounding task to a larger scale, covers more concepts and allows more flexible target regions.

1.3 Methods for Vision and Language

There is a significant literature on techniques for various language and vision tasks, with image captioning and visual question answering(VQA) being two of the most studied tasks. Modern captioning systems [63, 38, 115] produce descriptions by using encoder-decoder architectures, typically consisting of a convolutional network for encoding an image and a recurrent network for decoding a sentence. Techniques for VQA are based on a joint encoding of the image and the question to retrieve or generate an answer [108, 130, 61]. A criticism of the captioning task is that captions in existing datasets (*e.g.*, MS COCO dataset [70]) can be generated by identifying the dominant categories and relying on a language model. State-of-the-art systems are often matched by simple nearest-neighbor retrieval approaches [36, 133]. Visual question-answering systems [12] face a similar issue that most questions can be answered by relying on common-sense knowledge (*e.g.*, the sky is often blue). Some recent attempts have been made to address these issues [59]. The basic architectures for these tasks have been improved in a number of ways such as by incorporating attention mechanisms [76, 9, 119, 130, 43, 61] and improved language models [35, 102].

1.3.1 Approaches for grounding language to image regions

Techniques for localizing regions in an image given a natural language phrase can be broadly categorized into two groups: single-stage segmentation-based techniques and two-stage detection-and-ranking based techniques.

Single-stage methods [52, 73, 68, 104, 82, 126, 23, 125] predict a segmentation mask given a natural language phrase by leveraging techniques used in semantic segmentation. These methods condition a feed-forward segmentation network, such as a fully-convolutional network or U-Net, on the encoding of the natural language (*e.g.*, LSTM over words). The advantage is that these methods can be directly optimized for the segmentation performance and can easily handle stuff categories as well as different numbers of target regions. However, they are not as competitive on small-sized objects. We compare a strong baseline of RMI [73] on our dataset.

More state-of-the-art methods are based on a two-stage framework of region proposal and ranking. Significant innovations in techniques have been due to the improved techniques for object detection (*e.g.*, Mask R-CNN [46]) as well as language comprehension. Some earlier works [81, 129, 83, 53, 77, 97, 118, 75, 25, 91] adopt a joint image-language embedding model to rank object proposals according to their matching scores to the input expressions. More recent works improve the proposal generation [131, 25], introduce attention mechanisms [33, 126, 7] for accurate grounding, or leverage weak supervision from captions [122, 31].

The two-stage framework has also been further extended to modular comprehension inspired by neural module networks [11]. For example, Hu *et al.* [51] introduce a compositional modular network for better handling of attributes and relationships. Yu *et al.* [128] propose a modular attention network (MattNet) to factorize the referring task into separate ones for the noun phrase, location, and relationships. Liu *et al.* [76] improves MattNet by removing easy and dominant words and regions to learn more challenging alignments. Several recent works [132, 119, 124, 74, 13, 37, 14] also apply reasoning on graphs or trees

for more complicated phrases. These approaches have several appealing properties such as more detailed modeling of different aspects of language descriptions. However, these techniques have been primarily evaluated on datasets with a closed set of categories, and often with ground-truth instances provided.

Sadhu *et al.* [100] proposes zero-shot grounding to handle phrases with unseen nouns. Our work in Chapter 4 emphasizes further on the large number of categories, attributes and relationships, providing supervision over these long-tailed concepts and more detailed and straightforward evaluation.

1.3.2 Large-scale pre-training of image and text representations

With the availability of vast numbers of paired image and text data on the Internet, the development of image and text encoders, and the growth of computing resources, it now becomes possible to train large-scale representation learning models for jointly understanding images and text. These models can be applied to various cross-modal tasks such as zero-shot classification, image-text retrieval, visual question answering, action recognition in videos, geo-localization and so forth.

CLIP [94] trains an image encoder and a text encoder jointly on 400 million image-text pairs from the Internet: given an image, the task is to predict, among a sampled set of text descriptions, which one is paired with the input image in the training data. Radford *et al.* demonstrate the quality of image representations through training linear classifiers on top of image embeddings. They have also applied the encoders in downstream tasks such as geo-localization, optical character recognition, facial emotion recognition, and action recognition. [42] applies CLIP as a guidance for image and caption generation models. In Chapter 5 we select CLIP as an example to analyze the effectiveness of pre-trained models. While the analysis in [94] is more focused on image categorization based on the category name alone, we look further into CLIP’s capability of understanding adjectives or attribute phrases in fine-grained domains.

ALIGN [58] leverages a noisy dataset of over one billion image-text pairs, obtained without expensive filtering or post-processing steps in the Conceptual Captions dataset. It shows that the scale of data can make up its noise and reach state-of-the-art performance on various tasks.

UNITIER [27] applies a transformer on top of image and text encoders to better align image regions with words. They leverage image-text pairs from four image captioning datasets COCO, Visual Genome, Conceptual Captions [103], and SBU Captions [86], and train the model on four pre-training tasks: masked language modeling conditioned on image, masked region modeling conditioned on text, image-text matching, and word-region alignment.

WenLan [54] constructs a Chinese image-text paired dataset containing 30 million pairs and applies a two-tower structure on top of the image and text encoders for better contrastive learning. It was shown to outperform [94] and [27] on various downstream tasks.

CHAPTER 2

DESCRIBING FINE-GRAINED CATEGORIES

Attribute-based representations have been used for describing instances within a basic-level category as they often share a set of high-level properties. These attributes serve as basis for human-centric tasks such as retrieval and categorization [117, 64, 88], and for generalization to new categories based on a description of their attributes [40, 41, 99, 67]. However, most prior work has relied on a fixed set of attributes designed by experts. This limits their scalability to new domains since collecting expert annotations are expensive, and results in models that are less robust to noisy open-ended descriptions provided by a non-expert user.

Instead of discrete attributes, we investigate the use of *attribute phrases* for describing instances. Attribute phrases are short sentences that describe a unique semantic visual property of an object (*e.g.*, “red and white color”, “wing near the top”). Like captions, they can describe properties in a compositional manner, but are typically shorter and only capture a single aspect. Like attributes, they are modular, and can be combined in different ways to describe instances within a category. Their compositionality allows the expression of large number of properties in a compact manner. For example, colors of objects, or their parts, can be expressed by combining color terms (*e.g.*, “red and white”, “green and blue”, *etc.*). A collection of these phrases constitutes the semantic space of describable attributes and can be used as a basis for communication between a human and computer for various tasks.

We begin by collecting a dataset of attribute phrases by asking annotators to describe five visual differences between random pairs of airplanes from the OID airplane dataset [112].



Figure 2.1: **Reference games with attribute phrases.** *Left:* Each annotation in our dataset consists of five pairs of attribute phrases. *Right:* A *reference game* played between a speaker who describes an attribute of an image within a pair and a listener whose goal is to pick the right one.

Each difference is of the form “ P_1 vs. P_2 ” with phrases P_1 and P_2 corresponding to the properties of the left and right image respectively (Figure 2.1). By collecting multiple properties at a time we obtain a diverse set of describable attributes. Moreover, phrases collected in a contrastive manner reveal attributes that are better suited for fine-grained discrimination. The two phrases in a comparison describe the same underlying attribute (*e.g.*, *round nose* and *pointy nose* both describe the shape), and reflect an axis of comparison in the underlying semantic space. We then analyze the ability of automatic methods to generate these attribute phrases using the collected dataset. In particular we learn to generate descriptions and ground them in images in the context of a *reference game* (RG) between a *speaker* S and a *listener* L (Figure 2.1). S is provided with a pair of images $\{I_1, I_2\}$ and produces a visual difference of the form P_1 (or “ P_1 vs. P_2 ”). L ’s goal is to identify which of the two images corresponds to P_1 . Reference games have been widely used to collect datasets describing objects within a scene. This work employs the framework to generate and reason about compositional language-based attributes for fine-grained visual categorization.

Our experiments show that a speaker trained to describe visual differences displays remarkable pragmatic behavior allowing a neural listener to rank the correct image with **91.4%** *top-5* accuracy in the RG compared with **80.6%** of a speaker trained to generate

captions non-contrastively. We also investigate a family of *pragmatic speakers* who generate descriptions by jointly reasoning about the listener’s ability to interpret them, based on the work of Andreas and Klein [10]. Contrastively trained pragmatic speakers offer significant benefits (on average **7%** higher *top-5* accuracy in RG across listeners) over simple pragmatic speakers. The resulting speakers can be used to generate attribute-based explanations for differences between two categories. Moreover, given a set of attribute phrases, the score of an image with respect to each phrase according to a listener provides a natural embedding of the image into the space of semantic attributes. On the task of image classification on the *FGVC aircraft dataset* [80] this representation outperforms existing attribute-based representations by **20%** accuracy.

For the task of fine-grained recognition, the work of Reed *et al.* [96] is the most related to ours. They ask annotators on Amazon Mechanical Turk to describe properties of birds and flowers, and use the data to train models of images and text. They show the utility of such models for zero-shot recognition where a description of a novel category is provided as supervision, and for text-based image retrieval. Another recent work [113] showed that referring expressions for images within a set can be generated simply by enforcing separation of image probabilities during decoding using beam search. However, their model was trained on context agnostic captions. Our work takes a different approach. First, we collect attribute phrases in a contrastive manner that encourages pragmatic behavior among annotators. Second, we ask annotators to provide multiple attribute descriptions, which as we described earlier, allows modular reuse across instances, and serves as an intermediate representation for various tasks. Attribute phrases capture the spectrum between basic attributes and detailed captions. Like “visual phrases” [99] they capture frequently occurring relations between basic attributes.

2.1 A Dataset of Attribute Phrases

We rely on human annotators to discover the space of descriptive attributes. Our annotations are collected on images from the OID aircraft dataset [112]. The annotations are organized into 4700 image pairs (1851 images) in training set, 2350 pairs (1730 images) in validation set, and 2350 pairs (2705 images) in test set. Each pair is chosen by picking two different images uniformly at random within the provided split in the OID aircraft dataset.

Annotators from Amazon Mechanical Turk are asked to describe five properties in the form “ P_1 vs. P_2 ”, each corresponding to a different aspect of the objects in the left and the right image respectively. We also provide some examples as guidance to the annotators. The interface shown in Figure 2.2 is lightweight and allows rapid deployment compared to existing approaches for collecting attribute annotations where an expert decides the set and semantics of attributes ahead of time. However, the resulting annotations are noisier and reflect the diversity of open-ended language-based descriptions. A second pass over the data is done to check for consistency, after which about 15% of the description pairs were discarded.

Figure 2.1 shows an example of our dataset. Annotations describe the shapes of parts (nose, wings and tail), relative sizes, orientation, colors, types of engines, *etc.* Most descriptions are short with an average length of 2.4 words on each side, although about 4.3% of them have more than 4 words. These are qualitatively different from image captions which are typically longer and more grammatical. However, each annotation provides five different attribute pairs.


The OID dataset also comes with a set of expert-designed attributes. A comparison with OID attributes shows that attribute phrases capture novel properties that describe the relative arrangement of parts (*e.g.*, “door above the wing”, “wing on top”), color combinations, relative sizes, shape, and number of parts (*e.g.*, “big nose”, “more windows”, *etc.*) Section 2.3.4 shows a visualization of the space of attribute phrases. Section 2.3.3 pro-

Describe differences between the two aeroplane images

Instructions:

- Annotate each one of the three tasks
- Press *Next* to move to the next pair and *Submit* once done.
- If the images do not display, your browser may not support this interface. Try the latest Chrome, Safari or Firefox browsers.

[Click here](#) to see example answers before you start.



List 5 differences between the two images

1.	<input type="text"/>	VS	<input type="text"/>
2.	<input type="text"/>	VS	<input type="text"/>
3.	<input type="text"/>	VS	<input type="text"/>
4.	<input type="text"/>	VS	<input type="text"/>
5.	<input type="text"/>	VS	<input type="text"/>

pair 1 of 3
[Previous](#) | [Next](#)

Figure 2.2: **The interface used to collect five different *attribute phrase* pairs adapted from [78].** Amazon Mechanical Turkers were paid \$0.12 for annotating three pairs.

vides a direct comparison of OID attributes and those derived from our data on the task of FGVC-aircraft variant classification [80].

2.2 Models

2.2.1 Speaker models

A speaker maps visual inputs to attribute phrases. We consider two speakers; a *simple speaker* (SS) that takes a single image as input and produces a description, and a *discerning speaker* (DS) that takes two images as input and produces a single (or a pair of) description(s).

Both our speaker models are based on the show-and-tell model [115] developed for image captioning. Images are encoded using a convolutional network and decoded into a sentence using a recurrent network over words. We use one-hot encoding for 730 words with frequency greater than 5 in the training set. We consider f_c7 layer outputs of the

VGG-16 network [105] plus two fully-connected layers with ReLU units [84] on top as the image feature, and a LSTM model [48] with 2048 hidden units to generate the sentences. The image feature is fed into the LSTM not only as the initial input, but also in each state input together with word embeddings. This led to an improved speaker in our experiments. For the *discerning speaker*, we concatenate two image features as input to the LSTM. At test time we apply beam search with beam size 10 and get 10 output descriptions from each image (pair). Although the *discerning speaker* is trained to generate phrase pairs, we can simply take the first (or second) half of the pair and evaluate it in the same way as a *simple speaker*.

We also consider a *pragmatic speaker* that generates contrastive captions by reasoning about the listener’s ability to pick the correct image based on the description. Andreas and Klein [10] proposed a simple strategy to do so by reranking descriptions of an image based on a weighted combination of (a) *fluency* – the score assigned by the speaker, and (b) *accuracy* – the score assigned by the listener on the referred image. Various pragmatic speakers are possible based on the choice of speakers and listeners. The details are described in Section 2.3.2.

Optimization details: Our implementation is based on Tensorflow [5]. The descriptions are truncated at length 14 when training the LSTM. The VGG-16 network is initialized with weights pre-trained on ImageNet dataset [66]. We first fix the VGG-16 weights and train the rest of the network, using Adam optimizer [62] with initial learning rate 0.001, $\beta_1 = 0.7$, $\beta_2 = 0.999$ and $\epsilon = 1.0 \times 10^{-8}$. We have batch normalization [55] in fully connected layers after VGG-16, and drop out with rate 0.7 in LSTM. We use batch size 64 for 40000 steps (~ 28 epochs). Second, we fine tune the whole network with initial learning rate modified to 5×10^{-6} , batch size 32 for another 40000 steps.

2.2.2 Listener models

A listener interprets a single (or a pair of) attribute phrase(s), and picks an image within a pair by measuring the similarity between the phrase(s) and images in a common embedded space. Once again we consider two listeners: a *simple listener* (SL) that interprets a single phrase, and a *discerning listener* (DL) that interprets a phrase pair. The *simple listener* models the score of the image I_1 within a pair (I_1, I_2) for a phrase P as:

$$p(I_1|P) = \sigma(\phi(I_1)^T\theta(P), \phi(I_2)^T\theta(P)).$$

Here ϕ and θ are embeddings of the image and the phrase respectively, and σ is the softmax function $\sigma(x, y) = \exp(x)/(\exp(x) + \exp(y))$. Similarly, a *discerning listener* models the score of an image by comparing it with an embedding of the phrase pair $\theta([P_1 \text{ vs. } P_2])$. A simple way to construct a discerning listener from a simple listener is by averaging the predictions from the left and right phrases, *i.e.*, $p(I|[P_1 \text{ vs. } P_2]) = (p(I|P_1) + p(I|P_2)) / 2$.

We follow the setup of the speaker to embed phrases and use the final state of a LSTM with 1024 hidden nodes as the phrase embedding. The vocabulary of words is kept identical. For image features, once again we use the fc7 layer of the VGG-16 network and add a fully-connected layer with 1024 units and ReLU activation. The parameters are learned by minimizing the cross-entropy loss.

We also evaluate two variants of the *simple listener*, SL_r and SL, based on whether it is trained on non-contrastive data (I_1, I_2, P_1) where I_2 is a *random image* within the training set, or the contrastive data where I_2 is the other image in the annotation pair.

Optimization details: We first fix the VGG-16 network and use Adam optimizer with initial learning rate = 0.001, $\beta_1 = 0.7$, batch size = 32 for 2000 steps (4000 steps for SL_r model), then fine-tune the entire model with a learning rate 1×10^{-5} for another 7000-10000 steps.

Human listener. We also consider human annotators to perform the task of the listener in the RG. For each generated phrase that describes one image out of an image pair, we let three users to pick which image out of the pair the phrase is referring to. However, unlike (most) human speakers, neural speakers can produce irrelevant descriptions. Thus, in addition to the choice of left and right image, users have the option to say “*not sure*” when the description is ambiguous. If two or more users out of three picked the same image, we say the human listener is certain about the choice, otherwise we say the human listener is uncertain.

2.3 Results

We evaluate various listeners and speakers on the dataset we collected in terms of their accuracy in the RG in Section 2.3.1 and Section 2.3.2 respectively. We then evaluate their effectiveness on a fine-grained classification task in Section 2.3.3, visualize the space of attribute phrases discovered from the data in Section 2.3.4, for text-based image retrieval in Section 2.3.5, and for generating visual explanations for differences between categories in Section 2.3.6.

2.3.1 Evaluating listeners

We first evaluate various listeners on human-generated phrases. For simple listeners, each annotation provides ten different reference tasks $(I_1, I_2, P) \rightarrow \{0,1\}$ corresponding to five different left and right attribute phrases. Each task is evaluated independently and accuracy is measured as the fraction of correct references made by the listener. Similarly, discerning listeners are evaluated by replacing P with “ P_1 vs. P_2 ” or “ P_2 vs. P_1 ”.

Accuracy using human speakers. The results are shown in Table 2.1. Training on contrastive data improves the accuracy of the simple listener slightly from 84.2% (SL_r) to 86.3% (SL) on the test set. Discerning listeners see both phrases at once and naturally perform better. There is almost no difference between a discerning listener that combines

Input	Speaker	Listener	Val	Test
P ₁	Human	SL _r	82.7	84.2
		SL	85.3	86.3
P ₁ vs. P ₂	Human	DL	88.7	88.9
		2×SL	89.6	89.3

Table 2.1: Accuracy (%) of various listeners in the RG using attribute phrases provided by a human speaker.

	1	2	3	4	5
Val	91.3	86.6	84.1	82.5	82.3
Test	92.3	87.4	85.9	84.0	81.6

Table 2.2: Accuracy (%) of the simple listener (SL) on RG using human-generated attribute phrases at positions one through five across the validation and test set. The accuracy decreases monotonically from one to five suggesting that the top attribute phrases are easier to discriminate.

two simple listeners by averaging their predictions (2×SL), and one that interprets the two phrases at once (DL). The results indicate that on our dataset the listener’s task is relatively easy and contrastive data does not provide any significant benefits. As a reference the accuracy of a human listener is close to 100% on human-generated phrases.

Are the top attributes more salient? As annotators are asked to describe five different attributes they might pick ones that are more salient first. We evaluate this hypothesis by measuring the accuracy of the listener (SL) on phrases as a function of the position of the annotation in the interface ranging from one for the top attribute to five for the last one. The results are shown in Table 2.2. The accuracy decreases monotonically from one to five suggesting that the first attribute phrase is easier for the listener to discriminate. We are uncertain if this is because the attributes near the top are more discriminative, or because the listener is better at interpreting these as they are likely to be more frequent in the training data. Nevertheless, attribute saliency is a signal we did not model explicitly and may be used to train better speakers and listeners (*e.g.*, see Turakhia and Parikh [109]).

2.3.2 Evaluating speakers

We use simple listeners, SL and SL_r , and the human listener to evaluate speakers. As described in Section 2.2.1 we use beam search to generate 10 descriptions for each image pair and evaluate them individually using various listeners. The discerning speaker generates phrase pairs but we simply take the first and second half separated by “vs.”, a special word in the vocabulary, and evaluate it using a simple listener (that sees only one phrase). If the word “vs.” is missing in the generated output we simply consider the entire sentence as the P_1 . Only 1 out of 23500 phrase pairs did not contain the “vs.” token.

For evaluation with humans we collect three independent annotations on a subset of 100 image pairs (with 10 descriptions each) out of the full test set. The listeners are considered to be correct when the probability of the correct image is greater than half. For human listener, we report the accuracy of when there is a majority agreement on the correct image, *i.e.*, when two or more users picked the correct image. For direct comparison with the simple speaker models, we also report the human listener accuracy when they are allowed to guess. This is the sum of earlier accuracy, and half of the cases when there is no majority agreement. Human annotators are uncertain when the generated descriptions are not fluent or when they are not discriminative. Therefore, a better human accuracy reflects speaker quality both in terms of fluency and discriminativeness. Some examples of the generated attribute phrases using various speakers are shown in Figure 2.3.

Accuracy of various speakers and listeners. Results on the full test set (Test) and the human-evaluated subset (Test*) are shown in Table 2.3. The accuracy of discerning speaker exceeds that of simple speaker by more than 10% no matter which listener to use. This result suggests that data collected contrastively using our annotation task allows direct training of speaker models that show remarkable context-sensitive behavior. Somewhat surprisingly we also see that the simple listeners are more accurate than the human listener when evaluated on descriptions generated by our speaker models. This is because humans tend to be more cautious in the reference game. For example, simple listeners will accept



Ground Truth:

- 1) small size **VS** large size
- 2) single seat **VS** more seated
- 3) facing left **VS** facing right
- 4) private **VS** commercial
- 5) wings at the top **VS** wings at the bottom

DS:

- 1) private plane **VS** commercial plane (p=0.3338)
- 2) private **VS** commercial (p=0.1648)
- 3) small plane **VS** large plane (p=0.0701)
- 4) facing left **VS** facing right (p=0.0355)
- 5) short **VS** long (p=0.0250)
- 6) white **VS** red (p=0.0228)
- 7) high wing **VS** low wing (p=0.0184)
- 8) small **VS** large (p=0.01775)
- 9) glider **VS** jetliner (p=0.0170)
- 10) white and blue color **VS** white red and blue color (p=0.0159)

SS:

- 1) no engine (p=0.2963)
- 2) small (p=0.1800)
- 3) private plane (p=0.0650)
- 4) on the ground (p=0.0519)
- 5) propellor engine (p=0.0322)
- 6) on ground (p=0.0250)
- 7) glider (p=0.0228)
- 8) white color (p=0.0163)
- 9) small plane (p=0.0151)
- 10) no propeller (p=0.0124)

Figure 2.3: **Example output of simple speaker SS and discerning speaker DS.** Simple speaker takes the left image in the green box as input, while the discerning speaker takes both images as input. In brackets are the probabilities according to the speaker.

yellowish grass being referred to as “concrete” compared to green grass, but humans tend to view it as an unclear reference.

Does pragmatics help? Given that our discerning speaker can generate highly accurate contrastive descriptions, we investigate if additional benefits can be achieved if the speaker jointly reasons about the listener’s ability to interpret the descriptions. We employ the *pragmatic speaker* model of Andreas and Klein [10] where a simple speaker generates descriptions that are reranked by a simple listener using a weighted combination of speaker and listener scores. In particular, we rerank the output 10 sentences from speakers by the probabilities from simple listeners. We combine the listener probability p_l and speaker beam-search probability p_s as $p = p_s^\lambda \cdot p_l^{(1-\lambda)}$, and pick the optimal λ on a validation set annotated by a human listener. We found that the optimal λ is close to 0, so we decided to use p_l only for reranking on test set.

In Table 2.4, we report the accuracy of top k sentences ($k = 1, 5, 7$) of the human listener and the results after reranking on the Test* set. When using the listener score from

		Accuracy (%)				
		SL _r		SL		Human
	Top	Test*	Test	Test*	Test	Test*
SS	1	84.0	79.8	83.0	81.7	68.0 (77.0)
	5	80.0	79.2	78.0	80.6	64.2 (74.1)
	10	78.0	78.9	76.6	80.0	61.6 (72.4)
DS	1	94.0	92.8	92.0	92.8	82.0 (88.5)
	5	91.2	90.3	91.2	91.4	80.2 (86.7)
	10	88.6	88.8	90.0	90.5	77.9 (85.0)

Table 2.3: **Accuracy in the RG using different speakers and listeners.** Test represents the full test set consisting of 2350 image pairs. Test* represents a subset of 100 test set image pairs for which we collected human listener results. For the human listener, we report the accuracy when there is a majority agreement, and accuracy with guessing (in brackets). DS is significantly better at generating discriminative attribute phrases than SS.

SL_r, the average accuracy of the top five generated descriptions after reranking improves dramatically from 64.2% to 82.6% for the simple speaker. The accuracy of the discerning speaker also improves to 90%. This suggests that better pragmatics can be achieved if both the speaker and listener are trained in a contrastive manner. Surprisingly the contrastively-trained simple listener SL is less effective at reranking than SL_r. We believe this is because the SL overfits on the human speaker descriptions and is less effective when used with neural speakers.

Figure 2.4 shows an example pair and the output of different speakers. Simple speaker suffers from generating descriptions that are true to the target image, but fail to differentiate two images. Discerning speaker can mostly avoid this mistake. Reranking by listeners can move better sentences to the top and improves the quality of top sentences.

2.3.3 Fine-grained classification with attributes

We compare the effectiveness of attribute phrases to existing attributes in the OID dataset on the task of fine-grained classification on the FGVC aircraft dataset [80]. The OID dataset is designed with attributes in mind and has long-tail distribution over aircraft variants with 2728 models, while the FGVC dataset is designed for fine-grained classifica-

Human listener accuracy (%)				
Reranker listener				
	Top	None	SL _r	SL
SS	1	68.0 (77.0)	94.0 (96.0)	87.0 (92.0)
	5	64.2 (74.1)	82.6 (88.3)	80.8 (87.1)
	7	63.1 (72.8)	74.3 (82.0)	74.3 (82.4)
DS	1	82.0 (88.5)	95.0 (96.5)	95.0 (97.0)
	5	80.2 (86.7)	90.0 (93.3)	88.6 (92.8)
	7	79.1 (85.6)	86.7 (91.5)	86.1 (91.1)

Table 2.4: **Accuracy of pragmatic speakers with human listeners on the Test* set.** After generating the descriptions by the speaker model (either SS or DS), we use the listener model (SL_r or SL) to rerank them. We report the accuracy based on human listener from the user study. We report both the accuracy when there is majority agreement, and accuracy with guessing (in brackets). Pragmatic speakers are strictly better than non-pragmatic ones.

tion task with 100 variants each with 100 images. Both datasets are based on the images from the `airliners.net` website and have a few overlapping images. We exclude the 169 images from the FGVC test set that appear in the OID training+validation set in our evaluation.

There are 49 attributes in the OID dataset organized into 14 categories. We exclude three attributes – two referring to the airline label and model, most of which have only one training examples per category, and another that is rare. We then trained linear classifiers to predict each attribute using the `fc7` layer feature of the VGG-16 network. Using the same features and trained classifiers, we construct a 46 dimensional embedding of the FGVC images into the space of OID attributes. The attribute classifiers based on the VGG-16 network features are fairly accurate (66% mean AP across attributes) and outperforms the Fisher vector baseline included in the OID dataset paper.

For the attribute phrase embeddings, we first obtain the K most frequent ones in our training set. Given an image I, we compute the score $\phi(I)^T\theta(P)$ for each phrase P from a listener as the embedding. For a fair comparison the image features are kept identical to the OID attribute classifiers. We also explore an *opponent attribute space*, where instead of top phrases we consider the top phrase pairs. Phrase pairs represent an axis of comparison,

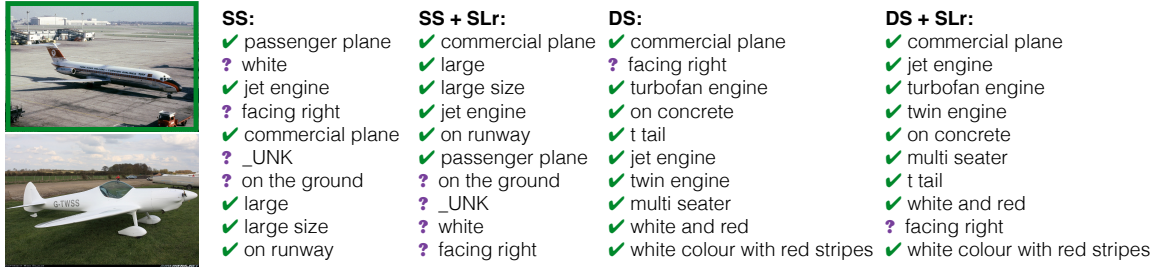


Figure 2.4: **An example output of various speakers.** Given the image pair, we use SS and DS to generate descriptions of the top left image. Outputs from SS and DS are listed in the order of probabilities from speaker beam search. Outputs of SS+SL_r and DS+SL_r are reranked by SL_r. Green checks mean human listener picks correct image with certain, while question marks mean human listener is uncertain which image is referred to. The results indicate that DS is better than SS, and reranking using listeners improves the quality of top sentences.

e.g., “small vs. medium”, or “red and blue vs. red and white”, and are better suited for describing relative attributes. We use the discerning listener for the embedding on the opponent attribute space.

Figure 2.5 shows a comparison of OID attributes and attribute phrases for various listeners and number of attributes. For the same number of attributes as the OID dataset, attribute phrases are **12%** better. With 300 attributes the accuracy improves to **32%**, about **20%** better than OID. These results indicate that attribute phrases provide a better coverage of the space of discriminative directions. The two simple listeners perform equally well and the opponent attribute space does not offer any additional benefits.

2.3.4 Visualizing the space of descriptive attributes

We visualize the space of the 500 most frequent phrases in the training set using the embedding of the simple listener model projected from 1024 dimensions to 2 using t-SNE [110] in Figure 2.6. Various semantically related phrases are clustered into groups. The cluster on the top right reflects color combinations; Phrases such as “less windows” and “small plane” are nearby (bottom right).

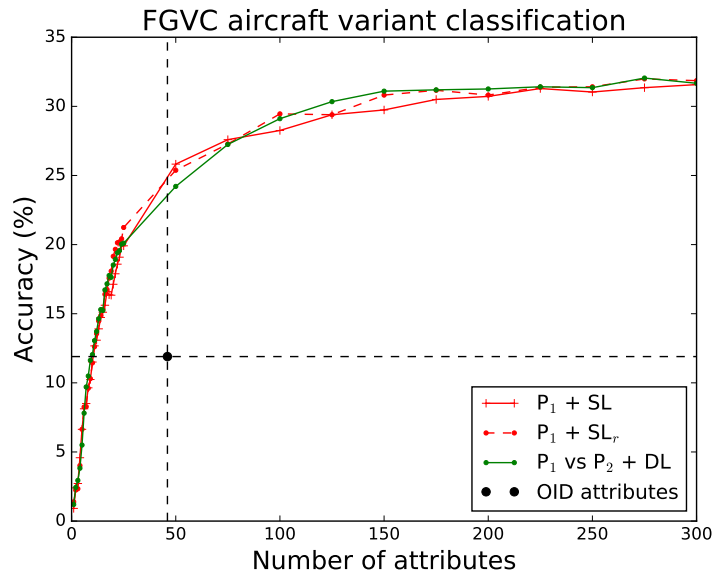


Figure 2.5: **Classification accuracy on FGVC aircraft dataset** using the 46 dimensional OID attributes and varying number of attribute phrases.

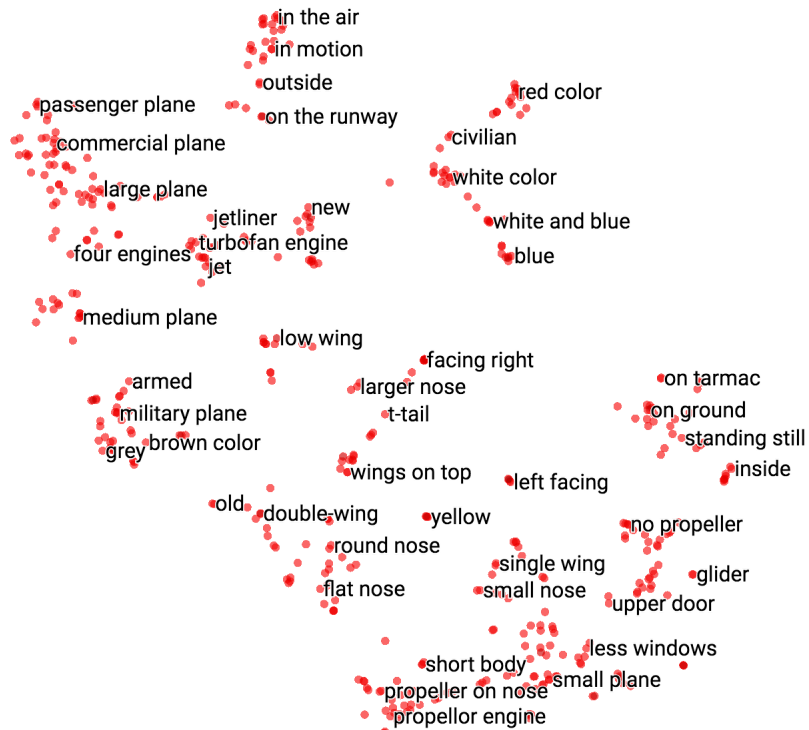


Figure 2.6: **Visualization of the 500 most frequent descriptions.** Each attribute is embedded into a 1024 dimensional space using the simple listener SL and projected into two dimensions using t-SNE [110].



Figure 2.7: **Top 18 images ranked by the listener for various attribute phrases as queries** (shown on top). We rank the images by the scores from the simple listener on the concatenation of the attribute phrases. The images are ordered from top to bottom, left to right.



Figure 2.8: **Top 10 discriminative attribute phrases for pairs of categories from FGVC aircraft dataset.** Descriptions are generated by the discerning speaker for each pair of images in the first and second category. The phrases sorted by the occurrence frequency provides an attribute-based explanation of the visual difference between two categories.

2.3.5 Image retrieval with descriptive attributes

The listeners also allows us to retrieve an image given one or more attribute phrases. Given a phrase P we rank the images in the test set by the listener scores $\phi(I)^T \theta(P)$. Figure 2.7 shows some query phrases and the 18 most similar images retrieved from the test set. These results were obtained by simply concatenating all the query phrases to obtain a single phrase. More sophisticated schemes for combining scores from individual phrase predictions are likely to improve results [101]. Our model can retrieve images with multiple attribute phrases well even though the composition of phrases does not appear in the training set. For example, “red and blue” only shows five times in total of 47,000 phrases in the training set, “pointy nose” and “on the runway” are never seen in a single phrase together.

2.3.6 Generating attribute explanations

The pairwise reasoning of a speaker can be extended to analyze an instance within a *set* by aggregating speaker utterances across all pairs that include the target. Similarly one can describe differences between two sets by considering all pairs of instances across the two sets. We use this to generate attribute-based explanations for visual differences between two categories. We select two categories A, B from FGVC aircraft dataset and randomly choose ten images from each category. For each image pair $(I_1 \in A, I_2 \in B)$, we generate ten phrase pairs using our discerning speaker. We then sort unique phrases primarily by their image frequency (number of images from target category described by the given description minus that from the opposite category), and when tied secondarily by their phrase frequency (number of occurrences of the phrase in target category minus that in the opposite category.) The top ten attribute phrases for the two categories for an example pair of categories are shown in Figure 2.8. The algorithm reveals several discriminative attributes between two such as “engine under wings” for 747-400, and “stabilizer on top of tail” for ATR-42.

2.4 Summary

We analyzed attribute phrases that emerge when annotators describe visual differences between instances within a subordinate category (airplanes), and showed that speakers and listeners trained on this data can be used for various human-centric tasks such as text-based retrieval and attribute-based explanations of visual differences between unseen categories. Our experiments indicate that pragmatic speakers that combine listeners and speakers are effective on the reference game [10], and speakers trained on contrastive data offers significant additional benefits. We also showed that attribute phrases are modular and can be used to embed images into an interpretable semantic space. The resulting attribute phrases are highly discriminative and outperform existing attributes on FGVC aircraft dataset on the fine-grained classification task.

CHAPTER 3

DESCRIBING TEXTURES

Texture is ubiquitous and provides useful cues for a wide range of visual recognition tasks. We rely on texture for estimating material properties of surfaces, for fine-grained discrimination of objects with a similar shape, for generating realistic imagery in computer graphics applications, *etc.* Texture is localized and more easily modeled than shapes that are affected by pose, viewpoint, or occlusion. The effectiveness of texture for perceptual tasks is also mimicked by deep networks trained on current computer vision datasets that have been shown to rely significantly on texture for discrimination (*e.g.*, [44, 29, 21, 50]).

While there has been significant work in the last few decades on visual representations of texture, limited work has been done on describing detailed properties of textures using natural language. The ability to describe texture in rich detail can enable applications on domains such as fashion and graphics, as well as to interpret discriminative attributes of visual categories within a fine-grained taxonomy (*e.g.*, species of birds or flowers) where texture cues play a key role. However, existing datasets of texture (*e.g.*, [28, 17]) are limited to a few binary attributes that describe patterns or materials, and do not describe detailed properties using the compositional nature of language (*e.g.*, descriptions of the color and shape of texture elements). At the same time, existing datasets of language and vision [12, 70, 103, 92, 60, 81, 129] primarily focus on objects and their relations with very limited treatment of textures. Addressing this gap in the literature, we introduce a new dataset containing rich natural language descriptions of textures called the Describable Textures in Detail Dataset (DTD²). It contains several descriptions of each image from the Describable Texture Dataset (DTD) [28] that are manually annotated. As seen in Figure 3.1, these

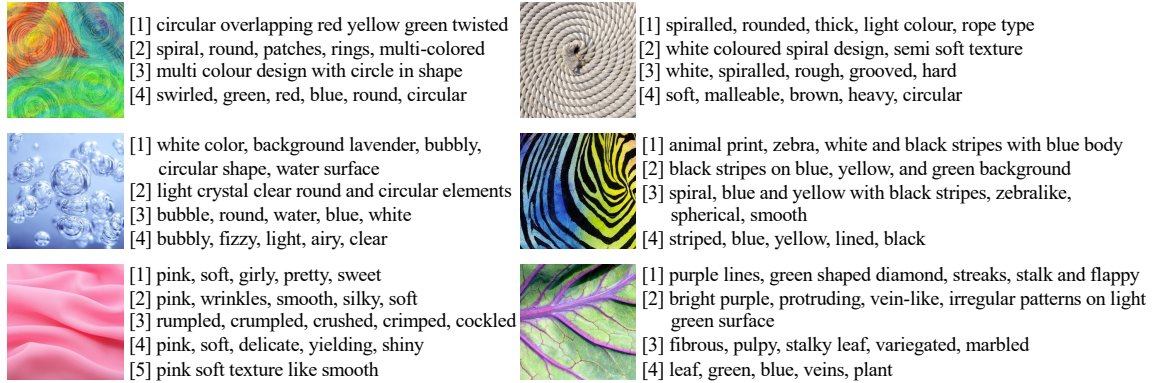


Figure 3.1: **We introduce the Describable Textures in Detail Dataset (DTD²)** consisting of texture images from DTD [28] with natural language descriptions, which provide rich and fine-grained supervision for various aspects of texture such as color compositions, shapes, and materials.

contain descriptions of colors of the structural elements (*e.g.*, “dots” and “lines”), their shape, and other high-level perceptual properties of texture (*e.g.*, “soft” or “protruding”). The resulting vocabulary vastly extends the 47 attributes present in the original DTD dataset (Section 3.1).

We argue that the domain of texture is rich and poses many challenges for compositional language modeling that are present in existing language and vision datasets describing objects and scenes. For example, to estimate the color of dots in a dotted texture the model must learn to associate the color to the dots and not to the background. Yet the domain of texture is simple enough that it allows us to analyze the robustness and generalization of existing vision and language models by synthetically generating variations of a texture. We conduct a systematic study of existing visual representations of texture, models of language, and methods for matching the two domains on this dataset (Sections 3.2, 3.3 and 3.3.3). We find that adopting pre-trained language models significantly improve generalization. However, they fail to capture detailed properties of texture which we critically analyze with synthetically generated variations of each texture by varying one attribute at a time (*e.g.*, foreground color and shape).

We also present two novel applications of our dataset (Section 3.4). First, we visualize what discriminative texture properties are learned by existing deep networks for fine-grained classification on natural domains such as birds, flowers, and butterflies. To this end, we generate “maximal images” for each category by “inverting” a state-of-the-art texture-based classifier [72] and describe these images using captioning models trained on DTD². We find that the resulting explanations are well aligned with the discriminative attributes of each category (*e.g.*, “Tiger Lily” flower is “black, red, white, and dotted” as seen in Figure 3.6-middle). We also show that models trained on DTD² offer improvements over expert-designed binary attributes on the Caltech-UCSD Birds dataset [116]. This complements the capabilities of existing datasets for explainable AI on these domains that focus on shapes, parts, and their attributes such as color. Texture provides a domain-independent, albeit incomplete way of describing interpretable discriminative properties.

In summary, our contributions are:

- A novel dataset of texture descriptions (Section 3.1).
- Evaluation of existing models of grounding natural language to texture (Section 3.2 and 3.3).
- Critical analysis of these models using synthetic, but realistic variations of textures with their descriptions (Section 3.3.3).
- Application of our models for describing discriminative texture attributes and building interpretable models on fine-grained domains (Section 3.4).

3.1 Dataset and Tasks

We begin by describing how we collected DTD² in Section 3.1.1, followed by the tasks and evaluation metrics in Section 3.1.2. DTD² contains multiple descriptions annotated by humans for each image in DTD. Each image I contains k descriptions $\mathbf{S} =$

$\{S_1, S_2, \dots, S_k\}$ from k different annotators who are asked to describe the texture presented in the image. Instead of a grammatically coherent sentence, we found it more effective and easier for them to list a set of properties separated by commas. Thus each description S can be interpreted as a set of phrases $\{P_1, P_2, \dots, P_n\}$. As seen in Figure 3.1, the ordering among phrases for a given description is somewhat arbitrary and in an initial experiment we found it hard to tell apart the description with original order of phrases from one with a random order, which motivates this annotation structure.

Figure 3.2 shows the statistics of the collected dataset. DTD² contains 5,369 images, 24,697 descriptions, and 22,435 unique phrases. We split the images into 60% training, 15% validation, and 25% test. Below we describe the details of the dataset collection pipeline and tasks. We will release the dataset publicly upon acceptance of the paper.

3.1.1 Dataset collection

Annotation We present each DTD image and its corresponding DTD texture category to 5 different Amazon Mechanical Turk workers, asking them to describe the texture using natural language with at least 5 words. Describable aspects of each image include texture, color, shape, pattern, style, and material (we provided examples of several texture categories in the annotation guidelines).

Verification After collecting the raw annotations, we manually verified all of them and removed annotations that were not appropriate or relevant for the image. For example, a breakfast waffle may have descriptions about the related food items such as strawberries instead of the description of texture which is our main goal. We also removed all images from “freckled” and “potholed” categories because they are primarily of human faces or scenes of roads. The resulting descriptions had few texture-related terms. We also excluded images with fewer than 3 valid descriptions.

Phrase retrieval Given an image, the goal is to rank phrases $p \in \mathcal{P}$ that are relevant to the image. Here \mathcal{P} is the set of all possible phrases, restricted to 655 frequent ones. For each image, the set of “true” relevant phrases are obtained by taking the union of phrases from all descriptions of the image. We can evaluate the ranked list according to several metrics:

- Mean Average Precision (MAP): area under the precision-recall curve;
- Mean Reciprocal Rank (MRR): One over the ranking of the first correct phrase;
- Precision at K (P@K): precision of the top K ranked phrases ($K \in \{5, 20\}$);
- Recall at K (R@K): recall of the top K ranked phrases ($K \in \{5, 20\}$).

Image retrieval from a phrase The task is to retrieve images given a query phrase. When taking phrases as the query, we consider all phrases $p \in \mathcal{P}$ as before and ask the retrieval model to rank all images in the test or validation set. The “true” list is all images that contain the phrase (in any of its descriptions). We consider the same metrics as the phrase retrieval task.

Image retrieval from a description When using descriptions the query, we consider all description $s \in \mathcal{S}$ as the input. Here \mathcal{S} is the set of all descriptions in the test or validation set. We ask the retrieval model to rank all images in the corresponding set. We evaluate the rank of the image from which the description was collected (MRR metric). This metric allows us to evaluate the compositional properties of texture over phrases (*e.g.*, “red dots” + “white background”). While we only quantitatively evaluate phrases and descriptions in the dataset, the ranking models can potentially generalize to novel descriptions or phrases over the seen words. We present qualitative results and a detailed study of the models in Section 3.3 and 3.3.3.

Description generation The task is to generate a description for an input image. Given each image I , we compare the generated description against the set of its collected descriptions $\{S_1, S_2, \dots, S_k\}$ using standard metrics for image captioning including BLEU-1,2,3,4 [87], METEOR [16], Rouge-L [69] and CIDEr [114]. However, we note that the task is open-ended and qualitative visualizations are just as important as these metrics.

3.2 Methods

We investigate three techniques learn the mapping between visual texture and natural languages on our dataset — a discriminative classification approach, a metric learning approach, and a language generation approach. They are explained in detail in the next three sections.

3.2.1 A discriminative classification approach

A simple baseline is to treat each phrase $p \in \mathcal{P}$ as a binary attribute and train a multi-label classifier to map the images to phrase labels. Given a texture image I , let $\psi(I)$ be an embedding computed using a deep network. We investigate activations from different layers of a ResNet101 [?] using mean-pooling over spatial locations as choices for the image embedding. For the classification task, we attach a classifier head h to map the embeddings to a 655-dimensional space corresponding to each phrase in our frequent set \mathcal{P} . The function h is modeled as a two-layer network – the first is fully-connected layer with 512 units with BatchNorm and ReLU activation; the second is a linear layer with 655 units followed by sigmoid activation. Given a training set of $\{(I_i, Y_i)\}_{i=1}^N$ where Y_i is the ground-truth binary labels across 655 classes for image I_i , the model is trained to minimize the binary cross-entropy loss: $L_{BCE} = \sum_i \ell_{bce}(h \circ \psi(I_i), Y_i)$, where $\ell_{bce}(y, z) = \sum_i (z_i \log(y_i) - (1 - z_i) \log(1 - y_i))$.

Training details The ResNet101 is initialized with weights pre-trained on ImageNet [34] and fine-tuned on our dataset. We consider features from layer-block 1 to 4 in the network

in our experiments. Each model is trained on the training split of our dataset for 75 epochs using the Adam optimizer [62] with an initial learning rate at 0.0001. We use 224×224 images for all our experiments. The hyper-parameters are set on the validation set.

Evaluation setup The classification scores over each phrase for each image are directly used to rank images or phrases for phrase retrieval or image retrieval with phrase input. Retrieving images given a description is more challenging since we need to aggregate the scores corresponding to different phrases, and the phrases in input descriptions may not be in \mathcal{P} . We found the following strategy works well: Given a description $S = \{P_1, P_2, \dots, P_n\}$ and an image I , obtain the scores for each phrase $s(P_i) = \sigma(h \circ \psi(I))_k$ where k is the index of the phrase $P_i \in \mathcal{P}$. If the phrase is not in the set, we consider all its sub-sequences that are present in \mathcal{P} and average the scores of them instead. For example, if the phrase “red maroon dot” is not present in \mathcal{P} , we consider all sub-sequences {red maroon, maroon dot, red, maroon, dot}, score each that is present in \mathcal{P} separately and then average the scores. By concatenating the top 5 phrases for an image we can also use the classifier to generate a description for an image. The key disadvantage of the classification baseline is that it treats each phrase independently, and does not have a natural way to score novel phrases (our baseline using sub-sequences is an attempt to handle this).

3.2.2 A metric learning approach

The metric learning approach aims to learn a common embedding over the images and phrases such that nearby image and phrase pairs in the embedding space are related. We adopt the standard metric learning approach based on triplet-loss [49]. Consider an embedding of an image $\phi(I)$ and of a phrase $\phi(P)$ in \mathbb{R}^d . Denote $\|\phi(I) - \phi(P)\|_2^2$ as the squared Euclidean distance between the two embeddings. Given an annotation (I, P) consisting of a positive (image, phrase) pair, we sample from the training set a negative image I' for P , and a negative phrase P' for I . We consider two losses; one from the negative phrase:

$$L_p(I, P, P') = \max(0, 1 + \|\psi(I) - \phi(P)\|_2^2 - \|\psi(I) - \phi(P')\|_2^2)$$

and another from the negative image:

$$L_i(P, I, I') = \max(0, 1 + \|\psi(I) - \phi(P)\|_2^2 - \|\psi(I') - \phi(P)\|_2^2)$$

The metric learning objective is to learn embeddings ψ and ϕ that minimize the loss $L = \mathbb{E}_{(I,P),(I',P')} (L_p + L_i)$ over the training set.

For embedding images we consider the same encoder as the classification approach with features from layer 2 and 4 from ResNet101. However, we add an additional linear layer with 256 units resulting in the embedding dimension $\psi(I) \in \mathbb{R}^{256}$. One advantage of the metric learning approach is that it allows us to consider richer embedding models for phrases. Specially we consider the following encoders:

- **Mean-pooling:** $\phi_{mean}(P) = \frac{1}{N_w} \sum_{w \in \text{tokenize}(P)} \text{embed}(w)$, where $\text{tokenize}(\cdot)$ splits the phrase into a list of words, $\text{embed}(\cdot)$ encodes each token into \mathbb{R}^{300} .
- **LSTM** [102]: $\phi_{lstm}(P) = \text{biLSTM}[\text{embed}(w) \text{ for } w \text{ in } \text{tokenize}(P)]$, with the same $\text{tokenize}(\cdot)$ and $\text{embed}(\cdot)$ as above. $\text{biLSTM}(\cdot)$ is a bi-directional LSTM with a single layer and hidden dimension 256 that returns the concatenation of the outputs on the last token from both directions.
- **ELMo** [90]: $\phi_{elmo}(P) = \text{ELMo}(P)$, where $\text{ELMo}(\cdot)$ uses pre-trained ELMo model [4] with its own tokenizer, and outputs the average embedding of all tokens in the phrase P .
- **BERT** [35]: $\phi_{bert}(P) = \text{BERT}(P)$, where $\text{BERT}(\cdot)$ uses pre-trained BERT model [3] with its own tokenizer, and outputs the average of last hidden states of all tokens in the phrase P .

To compute the final embedding of the phrase $\phi(P)$, we add a linear layer to the embeddings to a 256-dimensional space compatible with the image embeddings.

Training details We train this model on our training split using the Adam optimizer [62] with an initial learning rate at 0.0001. We find this model more prone to over-fitting than the classifier, therefore we apply an early stop mechanism when the image retrieval and phrase retrieval MAP on the validation set stops improving. Same as the classifier, ResNet101 is initialized with ImageNet [34] weights and fine-tuned on our data. $\text{embed}(\cdot)$ in ϕ_{mean} and ϕ_{lstm} is initialized with FastText embeddings [20, 1] and tuned end-to-end. Pre-trained ϕ_{elmo} and ϕ_{bert} are fixed in our training.

Evaluation setup Given the joint embedding space, one can retrieve phrases for each image and images for each phrase based on the Euclidean distance. Similar to the classifier we concatenate the top 5 retrieved phrases as a baseline description generation model. We also investigate a metric learning approach over descriptions rather than phrases where the positive and negative triplets are computed over (image, description) pairs. The language embedding models are the same since they can handle descriptions of arbitrary length.

3.2.3 A generative language approach

We adopt the Show-Attend-Tell model [123], a widely used model for image captioning. It combines a convolutional neural network to encode input images with an attention-based LSTM decoder to generate descriptions. Following the default setup, we encode images into the spatial features from the 4-th layer of ResNet101 (initialized with ImageNet [34] weights). The word embeddings are initialized from FastText [20, 1]. The entire model is then trained end-to-end on the training set, using the Adam optimizer [62] with initial learning rate 0.0001 for the image encoder and 0.0004 for the language decoder. We early stopping based on BLEU-4 score of generated descriptions on validation images.

This model is primarily used for the description generation task. In evaluation, we apply beam search of beam-size 5 and take the best description as the output.

Task:		Phrase Retrieval						Image Retrieval					
Data Split	Model	MAP	MRR	P@5	P@20	R@5	R@20	MAP	MRR	P@5	P@20	R@5	R@20
Validation	Classifier: Feat 1	13.10	37.20	16.05	10.68	4.94	13.04	10.64	25.06	11.57	9.37	6.13	17.78
	Classifier: Feat 2	17.65	44.91	22.41	14.59	6.85	17.60	13.00	29.24	14.60	11.08	8.52	22.54
	Classifier: Feat 3	26.43	60.52	32.47	20.71	9.93	25.00	15.62	31.79	17.28	13.34	9.42	28.52
	Classifier: Feat 4	26.51	59.24	33.07	20.84	10.07	25.16	15.85	33.06	17.83	13.02	9.94	27.28
	Classifier: Feat 1,4	25.78	58.28	31.58	20.31	9.55	24.44	15.85	32.35	18.35	13.51	10.24	28.03
	Classifier: Feat 2,4	26.57	59.19	32.65	21.11	9.99	25.50	16.19	32.53	17.47	13.56	10.63	28.69
Validation	Classifier: Feat 3,4	26.66	60.38	32.20	21.22	9.81	25.68	16.04	31.18	17.59	13.50	10.33	28.32
	Triplet: MeanPool	18.80	48.66	23.13	16.20	11.52	31.54	7.19	16.18	7.60	6.56	3.36	11.44
	Triplet: biLSTM	23.53	58.78	31.85	18.73	15.83	36.31	8.31	17.46	8.15	7.06	4.21	13.40
	Triplet: ELMo	28.13	68.46	37.02	21.11	18.44	41.12	11.25	24.05	12.79	10.27	5.85	18.57
Test	Triplet: BERT	31.68	72.59	40.67	22.96	20.23	44.50	15.22	31.39	16.27	12.56	9.07	25.69
	Classifier: Feat 2,4	27.12	61.28	33.50	21.71	16.07	41.48	14.75	33.94	18.75	16.02	6.47	19.32
	Triplet: BERT	31.77	74.12	41.70	23.60	20.17	45.04	13.50	31.12	16.52	14.57	5.24	17.32

Table 3.1: **Performance on phrase retrieval and image retrieval on DTD².** “Classifier: Feat x ” stands for the classifier with image features from ResNet layer block x (or a concatenation of two layers.) All triplet models in this table are trained with phrase input. Among the language models BERT works the best.

Model	MRR	Model	Bleu-1	Bleu-2	Bleu-3	Bleu-4	METEOR	Rouge-L	CIDEr
Classifier	12.40	Classifier: top 5	68.07	46.17	28.39	14.44	19.89	48.13	44.73
Triplet - phrase	12.92	Triplet: top 5	72.99	53.69	34.97	19.39	21.81	49.70	47.34
Triplet - description	13.95	Show-Attend-Tell	59.90	40.41	26.52	16.35	19.92	46.64	37.47

Table 3.2: **Retrieving textures from descriptions.**

Table 3.3: **Description generation on textures.** Synthesizing descriptions from phrases retrieved by the metric-learning based approach outperforms other baselines.

3.3 Experiments and Analysis

We present an analysis of the above models on the proposed tasks on DTD².

3.3.1 Phrase and image retrieval

Table 3.1 and 3.2 compare the classifier and the triplet model on phrase and image retrieval tasks as described in Section 3.1.2. Figure 3.3 and 3.4 show examples of the top 5 retrieved images and phrases.

We first compare the image features from different layers of ResNet with the classifier on the validation split, as shown in Table 3.1. Higher layer features perform better for phrase retrieval. For image retrieval, better performance is achieved with the combination



Figure 3.3: **Retrieve DTD² test images with language input.** We show top 5 retrieved images from the classifier, the triplet model with phrase input and with description input. From left to right we show example inputs of (1) phrases the classifier has been trained on, (2) novel phrases beyond the frequent phrase classes, and (3) full descriptions.

of features from different layers. We select to use the features from layer 2 and 4 for all classifiers and triplet models in subsequent experiments. Table 3.1 also compares language encoders on the triplet model. The performance of both phrase and image retrieval depends largely on the language encoder, and BERT performs the best. On the test set the trends are similar where the triplet model is better at phrase retrieval while the classifier is slightly better at image retrieval.

Table 3.2 shows results of image retrieval from descriptions and here too the triplet model outperforms the other two models. As shown in Figure 3.3-right, although the models trained on phrases work reasonably well, the triplet model trained on descriptions is able to model contextual information better.

3.3.2 Description generation

We compare the Show-Attend-Tell model [123] with a retrieval based approach. From the classifier or the triplet model we retrieve the top k phrases and concatenate them in the order of their score to form a description. As shown in Table 3.3, the triplet model reaches

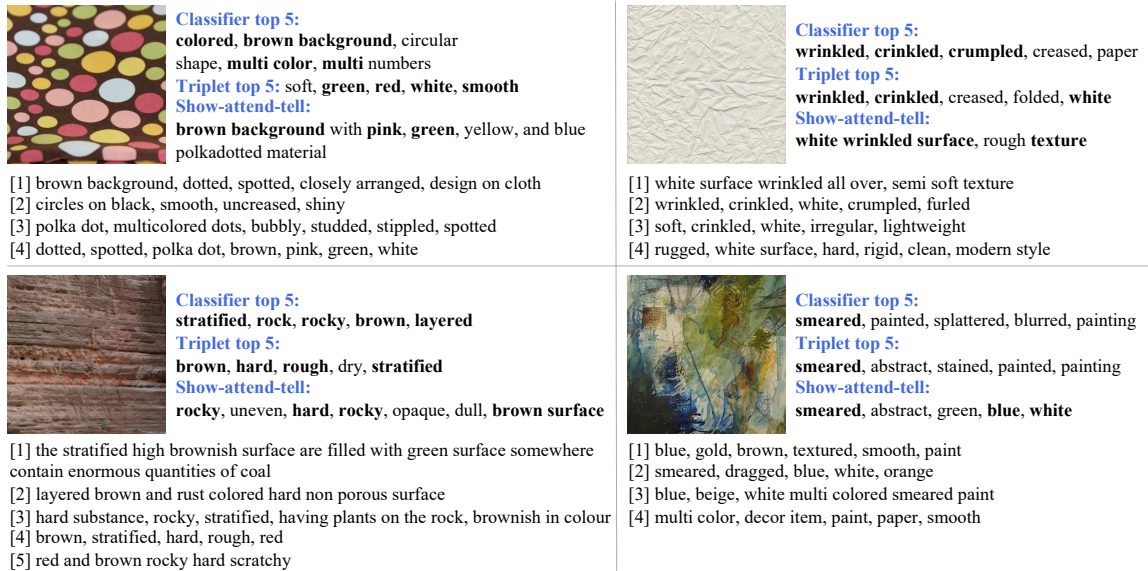


Figure 3.4: **Phrase retrieval and description generation on DTD² test images.** For each input image, we list ground-truth descriptions beneath, and generated descriptions on the right. For the classifier and the triplet model, we concatenate the top 5 retrieved phrases as the description. Bold words are the ones included in ground-truth descriptions.

Model	Foreground	Background	Color+Pattern	Two-colors
Classifier	45.45±20.34	59.82±9.63	35.95±21.48	26.82±14.17
Triplet - phrase	46.55±20.65	52.00±6.32	41.73±22.77	27.45±15.13
Triplet - description	47.64±18.97	53.64±4.66	35.77±21.12	21.59±13.77
Random guess	50.00	50.00	7.40	5.26

Table 3.4: **Image retrieval performance of R-Precision on synthetic tasks.**

higher scores on the metrics. However, notice that in Figure 3.4 the generative model’s descriptions are more fluent and covers both the color and pattern of the images, while the retrieval baselines (especially the classifier) repeat phrases with similar meanings.

3.3.3 A critical analysis of language modeling

In this section, we evaluate the proposed models on tasks where we systematically vary the distribution of underlying texture attributes. This is relatively easy to do for textures than natural images (*e.g.*, changing the color of dots) and allows us to understand the de-

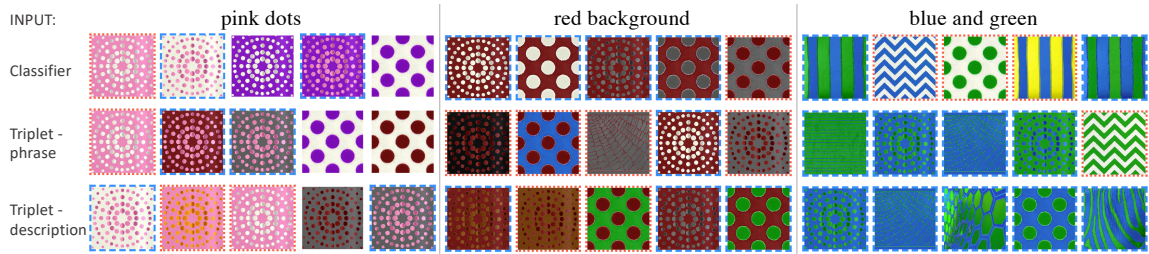


Figure 3.5: **Retrieval on synthetic images.** Positive images are in dashed blue borders, hard negative ones are in dotted red borders.

gree to which the models learn disentangled representations. We describe four tasks with varying degrees of difficulty to highlight the strengths and weaknesses of these models.

Automatically generating textures and their descriptions To systematically generate textures with descriptions, we follow this procedure:

- Take the 11 most frequent colors in DTD² (white, black, brown, green, blue, red, yellow, pink, orange, gray, purple) and set their RGB values manually.
- Take 10 common two-color images from ten different categories. We choose:
 - Type A: 5 images with “foreground on background”: [‘dots’, ‘polka-dots’, ‘swirls’, ‘web’, ‘lines’ (thin lines on piece of paper)], and
 - Type B: 5 images with no clear distinction between the foreground and background: [‘squares’ (checkered), ‘hexagon’, ‘stripes’ (zebra-like), ‘zigzagged’, ‘banded’ (bands with similar width)].
- For each of these 10 images, we manually extract masks for the foreground and background (Type A), or two foreground colors (Type B).
- For each of the 10 images, generate a new image by picking 2 different colors from the 11 and modify pixel values of the two regions using the corresponding RGB value. This results in $10 \times 11 \times 10 = 1,100$ images.

- For each synthetic image, we construct the ground-truth description as “[color1] [pattern], [color2] background”(such as “pink dots, white background”) for Type A, and “[color1] and [color2] [pattern]”(such as “yellow and gray squares”) for Type B.

Experiment 1: Foreground. On Type A set we construct:

- **Query:** A query of the form “[color=c] [pattern=p]” (*e.g.* “pink dots”).
- **Positive set:** [color=c] [pattern=p] on randomly colored background (*e.g.* “pink dots, white background”).
- **Negative set:** Randomly colored ($\neq c$) [pattern=p] on [color=c] background (*e.g.* “blue dots, pink background”).
- **Result:** Input the query description, we use the models to rank images from both the positive and negative set, and report R-Precision: the precision of top R predictions, where R is the number of positive images. The results are listed in Table 3.4 first column. Since half the images have the right attribute the chance performance is 50% and the various models are nearly at the chance level. Figure 3.5 shows that the model is unable to distinguish between “pink dots” and “dots on a pink background”. This illustrates that the models are unable to associate color correctly with the foreground shapes.

Experiment 2: Background. This is similar to Experiment 1 but we focus on the background instead. On Type A set we construct: we know the name of its pattern (such as “dots”, “squares”, selected from the more frequent phrases that matches the category) and names of two colors (color1 and color2).

- **Query:** A query “[color=c] background” (*e.g.* “pink background”).
- **Positive set:** Randomly colored pattern on [color=c] background (*e.g.* “red dots on pink background”).

- **Negative set:** Random pattern of [color=c] on any [color≠c] background (e.g. “pink dots on white background”).
- **Result:** R-precision is shown in Table 3.4 second column. Once again the chance performance is 50% and the various models are nearly at the chance level. Figure 3.5-middle shows that the model is unable to distinguish between “red background” and “red dots on random background”.

Experiment 3: Color+Pattern. On both Type A and B images we construct:

- **Query:** A query “[color=c] [pattern=p]” (e.g. “pink dots”).
- **Positive set:** [color=c] [pattern=p] on random colored background, or with another color (e.g. “pink dots, white background”, “pink and blue squares”).
- **Negative set:** [color=c] [pattern≠p] or [color≠c] [pattern=p]. In other words the negative set contains images with the correct pattern but wrong color or the wrong pattern with the right color (e.g., “red dots” or “pink stripes”). Similar patterns (e.g., “lines” vs. “banded”) are not considered negative.
- **Result:** The positive and negative set is unbalanced which results in a chance performance of 7.4%. The models presented in the earlier section are able to rank the correct color and pattern combinations ahead of the negative set and achieve a considerably higher performance.

Experiment 4: Two Colors. On both Type A and B images we construct:

- **Query:** A query “[color=c1] and [color=c2]” (e.g. “pink and green”).
- **Positive set:** [color=c1] of random pattern on [color=c2] background (e.g. “pink dots on green background”), or [color=c1] and [color=c2] of random pattern (e.g. “pink and green squares”).

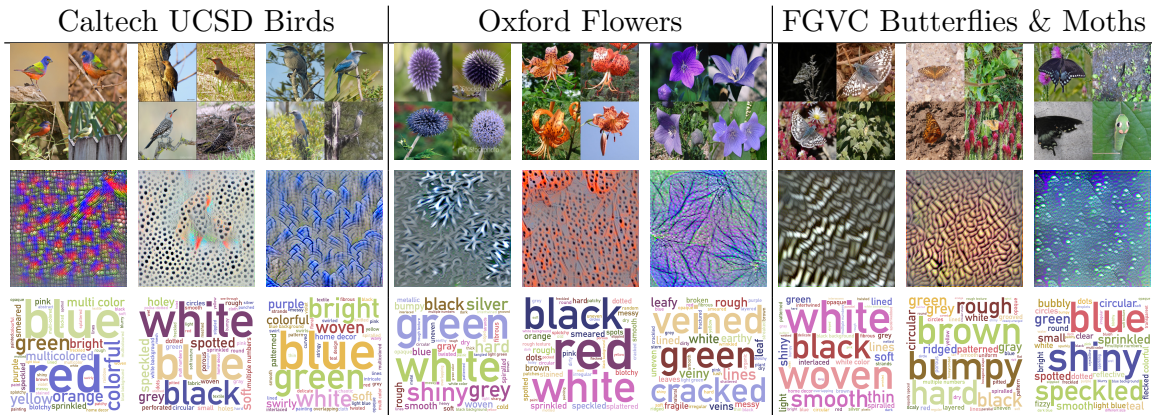


Figure 3.6: **Fine-grained categories visualized as their training images (top row), maximal texture images (middle row), and texture attributes (bottom row).** The size of each phrase in the cloud is inversely decided by its Euclidean distance to the input maximal texture image calculated by the triplet model.

- **Negative set:** pattern with one color from $\{c_1, c_2\}$ and another color $\neq \{c_1, c_2\}$ (e.g., “pink dots on yellow background”, “green and blue stripes”).
- **Result:** The positive and negative set are unbalanced which results in a chance performance of 5.26%. The models once again are able to rank the two color combinations ahead of the negative set and achieve a considerably higher performance. Figure 3.5-right shows an example.

Summary These experiments reveal that these models have some high-level discriminative abilities (Exp. 3, 4), but they fail to disentangle properties such as the color of the foreground elements from background (Exp. 1, 2). This leaves much room for improvement, motivating future work, such as those that enforces spatial agreement between the different attributes.

3.4 Applications

3.4.1 Describing textures of fine-grained categories

We analyze how the categories in fine-grained domains can be described by their texture. We consider categories from Caltech-UCSD Birds [116], Oxford flowers [85], and

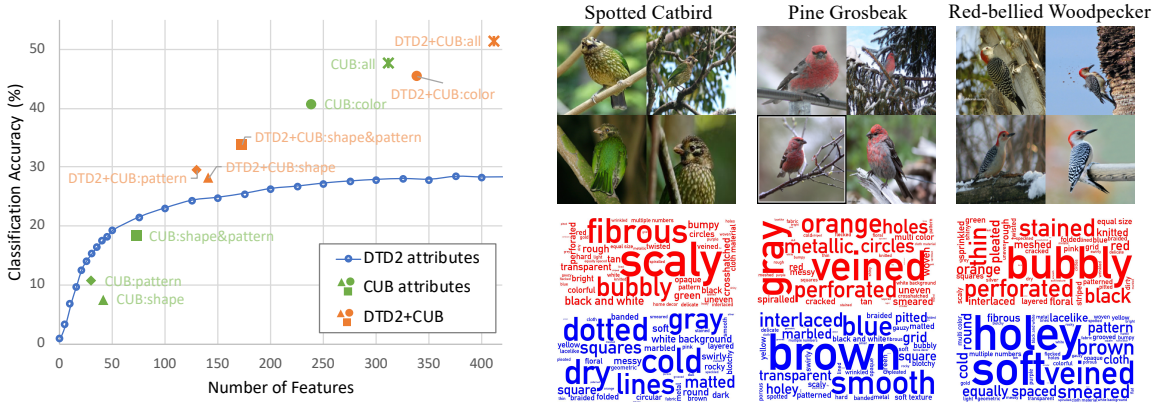


Figure 3.7: **Classification on CUB dataset with DTD² texture attributes.** *Left:* classification accuracy vs. number of input features. Orange and green markers with the same shape are comparable with the same set of CUB attributes with or without the DTD² attributes. *Right:* The phrase clouds display important phrases for a few bird categories. Red phrases correspond to positive weights and blue are negative for a linear classifier for the category. Font sizes represent the absolute value of the coefficient.

FGVC butterflies and moths [2] datasets. For each category, we follow the visualizing deep texture representations following [71] to generate the “maximal textures” — inputs that maximize the class probability using multi-layer bilinear CNN classifier [72]. These are provided as input to our triplet model (with BERT encoder and phrase input) trained on DTD² to retrieve the top phrases. Figure 3.6 shows several categories with their maximal textures together with a “phrase cloud” of the top retrieved phrases. These provide a qualitative description of each category.

3.4.2 Fine-grained classification with texture attributes

Here we apply models trained on our DTD² on the Caltech-UCSD birds dataset to show that embedding images into the space of texture attributes allows interpretable models for discriminative classification. Specifically, we input each image from the dataset to our phrase classifier (trained on DTD² and fixed) and obtain the log-likelihood over the 655 texture phrases as an embedding. We train a logistic regression model for the 200-way classification task. The dataset also comes with 312 binary attributes that describe the

shape, pattern and color of specific parts of a bird, such as “has tail shape squared tail”, “has breast pattern spotted”, “has wing color yellow”. There are 42 attributes for “shape”, 31 for “pattern” and 239 for “color”. We also train a logistic regression classifier on top of these attributes.

Figure 3.7 shows the performance by varying the number of texture phrases based on their frequency on DTD² as the blue curve. It also shows a comparison of bird-specific attributes with generic texture attributes learned on the DTD². Results using individual types of attributes are shown in green, while those using combinations are shown in orange. Texture attributes are able to distinguish bird species with a reasonable accuracy of 28.5%, outperforming CUB shape and pattern attributes. However, they do not outperform the part-based color attributes that are highly effective. Yet, combining class-specific attributes with texture lead to consistent improvements. On the right is visualization of discriminative texture attributes for some categories — we display phrases with the most positive weights in red, and those with the most negative weights in blue. These models provide a basis for interpretable explanations of discriminative features without requiring a category-specific vocabulary.

3.5 Summary

In conclusion, we presented a novel dataset of textures with natural language descriptions and analyzed the performance of several language and vision models. The domain of texture is challenging and existing models fail to learn a sufficiently disentangled representation leading to poor generalization on synthetic tasks. Yet, the learned models show some generalization capability to novel domains and enable us to provide interpretable models for describing the discriminative texture attributes in fine-grained domains. In particular they are complementary to existing domain-specific attributes on the CUB dataset.

CHAPTER 4

DESCRIBING REGIONS IN IMAGES

Existing efforts on grounding language descriptions to images have achieved promising results on datasets such as *Flickr30Entities* [93] and *Google Referring Expressions* [81]. These datasets, however, lack the scale and diversity of concepts that appear in real-world applications.

To bridge this gap we present the VGPHRASECUT dataset and an associated task of grounding natural language phrases to image regions called *PhraseCut* (Figure 4.1 and 4.2). Our dataset leverages the annotations in the *Visual Genome (VG)* dataset [65] to generate a large set of referring phrases for each image. For each phrase, we annotate the regions and instance-level bounding boxes that correspond to the phrase. Our dataset contains 77,262 images and 345,486 phrase-region pairs, with some examples shown in Figure 4.2. VG-PHRASECUT contains a significantly longer tail of concepts, which means there are more categories and attributes, and following an extremely imbalanced distribution. Unlike prior datasets that only focus on foreground objects, VGPHRASECUT has a unified treatment of not only object categories, which have well defined shapes such as people and cars, but also stuff, which are background regions with flexible shapes such as sky and grass. The phrases are structured into words that describe categories, attributes, and relationships, providing a systematic way of understanding the performance on individual cues as well as their combinations.

The *PhraseCut* task is to segment regions of an image given a *templated phrase*. As seen in Figure 4.1, this requires connecting natural language concepts to image regions. Our experiments shows that the task is challenging for state-of-the-art referring approaches

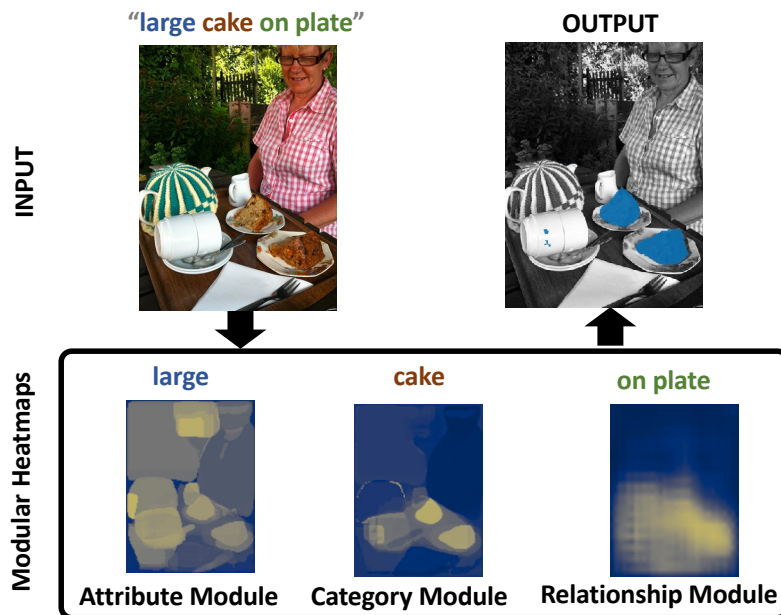


Figure 4.1: **PhraseCut task and our approach.** PhraseCut is the task of segmenting image regions given a natural language phrase. Each phrase is templated into words corresponding to *categories*, *attributes*, and *relationships*. Our approach combines these cues in a modular manner to estimate the final output.

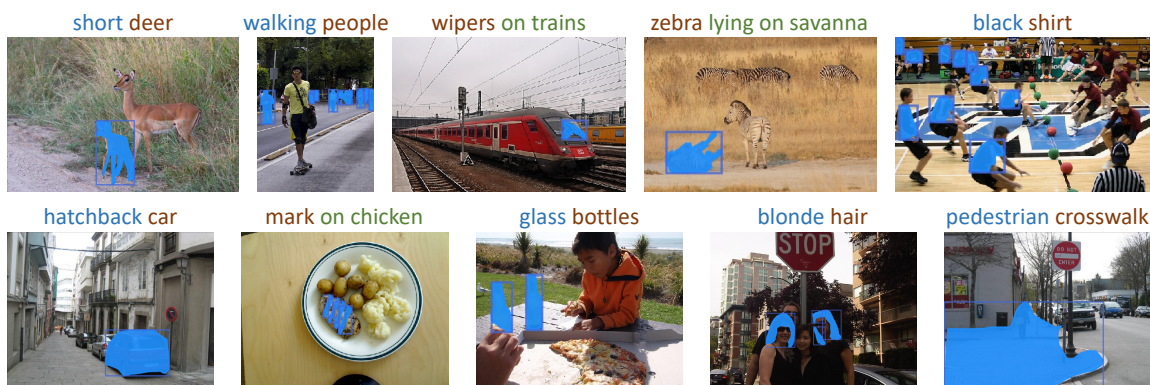


Figure 4.2: **Example annotations from the VGPHRASECUT dataset.** Colors (blue, red, green) of the input phrases correspond to words that indicate attributes, categories, and relationships respectively.

such as *MattNet* [128] and *RMI* [73]. We find that the overall performance is limited by the performance on rare categories and attributes. To address these challenges we present (i) a modular approach for combining visual cues related to categories, attributes, and relationships, and (ii) a systematic approach to improving the performance on rare categories and attributes by leveraging predictions on more frequent ones. Our category and attribute modules are based on detection models, whose instance-level scores are projected back to the image and further processed using an attention-based model driven by the query phrase. Finally, these are combined with relationship scores to estimate the segmentation mask (see Figure 4.1). Unlike existing two-stage methods (such as *MattNet* [128]) that outperform one-stage methods (such as *RMI* [73]) but fail to handle background stuff, our method processes objects and stuff categories in a unified manner. Our modular design, after the treatment of rare categories, outperforms existing end-to-end models trained on the same dataset.

Using the dataset we present a systematic analysis of the performance of the models on different subsets of the data. The main conclusions are: (i) object and attribute detection remains poor on rare and small-sized categories, (ii) for the task of image grounding, rare concepts benefit from related but frequent ones (*e.g.*, the concept “policeman” could be replaced by “man” if there were other distinguishing attributes such as the color of the shirt), and (iii) attributes and relationship models provide the most improvements on rare and small-sized categories. The performance on this dataset is far from perfect and should encourage better models of object detection and semantic segmentation in the computer vision community. The dataset and code is available at: <https://people.cs.umass.edu/~chenyun/phrasecut>.

4.1 The VGPHRASECUT Dataset

In this section, we describe how the VGPHRASECUT dataset was collected, the statistics of the final annotations, and the evaluation metrics. Our annotations are based on



Figure 4.3: **Illustrations of our VGPHRASECUT dataset collection pipeline.** **Step 1:** blue boxes are the sampling result; red boxes are ignored. **Step 2:** Phrase generation example in the previous image. **Step 3:** User interface for collecting region masks. **Step 4:** Example annotations from trusted and excluded annotators. **Step 5:** Instance label refinement examples. Blue boxes are final instance boxes, and red boxes are corresponding ones from Visual Genome annotations.

images and scene-graph annotations from the *Visual Genome (VG)* dataset. We briefly describe each step in the data-collection pipeline illustrated in Figure 4.3.

4.1.1 Data collection pipeline

Step 1: Box sampling Each image in VG dataset contains 35 boxes on average, but they are highly redundant. We sample an average of 5 boxes from each image in a stratified manner by avoiding boxes that are highly overlapping or are from a category that already has a high number of selected boxes. We also remove boxes that are less than 2% or greater than 90% of the image size.

Step 2: Phrase generation Each sampled box has *several* annotations of category names (e.g., “man” and “person”), attributes (e.g., “tall” and “standing”) and relationships with other entities in the image (e.g., “next to a tree” and “wearing a red shirt”). We generate one phrase for one box at a time, by adding categories, attributes and relationships that allow discrimination with respect to other VG boxes by the following set of heuristics:

1. We first examine if one of the provided categories of the selected box is unique. If so we add this to the phrase and tack on to it a randomly sampled attribute or relationship description of the box. The category name uniquely identifies the box in this image.

2. If the box is *not* unique in terms of any of its category names, we look for a unique attribute of the box that distinguishes it from boxes of the same category. If such an attribute exists we combine it with the category name as the generated phrase.
3. If *no* such an attribute exists, we look for a distinguishing relationship description (a relationship predicate plus a category name for the supporting object). If such a relationship exists we combine it with the category name as the generated phrase.
4. If all of the above fail, we combine all attributes and relationships on the target box and randomly choose a category from the provided list of categories for the box to formulate the phrase. In this case, the generated phrase is more likely to correspond to more than one instance within the image.

The attribute and relationship information may be missing if the original box does not have any, but there is always a category name for each box. Phrases generated in this manner tend to be concise but do not always refer to a unique instance in the image.

Step 3: Region annotation We present the images and generated phrases from the previous steps to human annotators on Amazon Mechanical Turk, and ask them to draw polygons around the regions that correspond to provided phrases. Around 10% of phrases are skipped by workers when the phrases are ambiguous.

Step 4: Automatic annotator verification Based on manual inspection over a subset of annotators, we design an automatic mechanism to identify trusted annotators based on the overall agreement of their annotations with the VG boxes. Only annotations from trusted annotators are included in our dataset. 9.27% phrase-region pairs are removed in this step.

Step 5: Automatic instance labeling As a final step we generate instance-level boxes and masks. In most cases, each polygon drawn by the annotators is considered an instance. It is further improved by a set of heuristics to merge multiple polygons into one instance and to split one polygon into several instances leveraging the phrase and VG boxes.

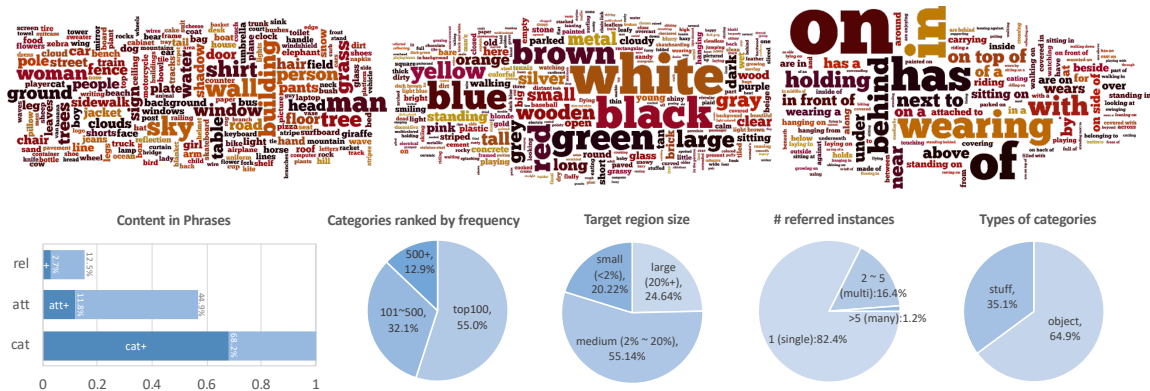


Figure 4.4: **Statistics of the VGPHRASECUT dataset.** *Top row:* Word clouds of categories (left), attributes (center), and relationship descriptions (right) in the dataset. The size of each phrase is proportional to the square root of its frequency in the dataset. *Bottom row:* breakdowns of the dataset into different subsets including contents in phrases (first), category frequency (second), size of walking target region relative to the image size (third), number of target instances per query phrase (fourth), and types of category (last). The leftmost bar chart shows the breakdown of phrases into those that have category annotation (cat) and those that can be distinguished by category information alone (cat+), and similarly for attributes and relationships.

4.1.2 Dataset statistics

Our final dataset consists of 345,486 phrases across 77,262 images. This roughly covers 70% of the images in Visual Genome. We split the dataset into 310,816 phrases (71,746 images) for training, 20,316 (2,971 images) for validation, and 14,354 (2,545 images) for testing. There is no overlap of COCO trainval images with our test split so that models pre-trained on COCO can be fairly used and evaluated. Figure 4.4 illustrates several statistics of the dataset. Our dataset contains 1,272 unique category phrases, 593 unique attribute phrases, and 126 relationship phrases with frequency over 20, as seen by the word clouds. Among the distribution of phrases (bottom left bar plot), one can see that 68.2% of the instances can be distinguished by category alone (*category+*), while 11.8% of phrases require some treatment of attributes to distinguish instances (*attributes+*). Object sizes and their frequency vary widely. While most annotations refer to a single instance, 17.6% of phrases refer to two or more instances. These aspects of the dataset make the *PhraseCut* task challenging.

4.1.3 Evaluation metrics

The *PhraseCut* task is to generate a binary segmentation of the input image given a referring phrase. We assume that the input phrase is parsed into attribute, category, and relationship descriptions. For evaluation we use the following intersection-over-union (IoU) metrics:

- cumulative IoU: $\text{cum-IoU} = (\sum_t I_t) / (\sum_t U_t)$, and
- mean IoU: $\text{mean-IoU} = \frac{1}{N} \sum_t I_t / U_t$.

Here t indexes over the phrase-region pairs in the evaluation set, I_t and U_t are the intersection and union area between predicted and ground-truth regions, and N is the size of the evaluation set. Notice that, unlike cum-IoU , mean-IoU averages the performance across all image-region pairs and thus balances the performance on small and large objects.

We also report the precision when each phrase-region task is considered correct if the IoU is above a threshold. We report results with IoU thresholds at 0.5, 0.7, 0.9 as $\text{Pr}@0.5$, $\text{Pr}@0.7$, $\text{Pr}@0.9$ respectively.

All these metrics can be computed on different subsets of the data to obtain a better understanding of the strengths and failure modes of the model.

4.2 A Modular Approach to PhraseCut

We propose **Hierarchical Modular Attention Network** (HULANet) for the PhraseCut task, as illustrated in Figure 4.5. The approach is based on two design principles. First, we design individual modules for category, attribute and relationship sub-phrases. Each module handles the long-tail distribution of concepts by learning to aggregate information across concepts using a module-specific attention mechanism. Second, instance-specific predictions are projected onto the image space and combined using an attention mechanism driven by the input phrase. This allows the model to handle stuff and object categories, as well as multiple instances in a unified manner. Details of each module are described next.

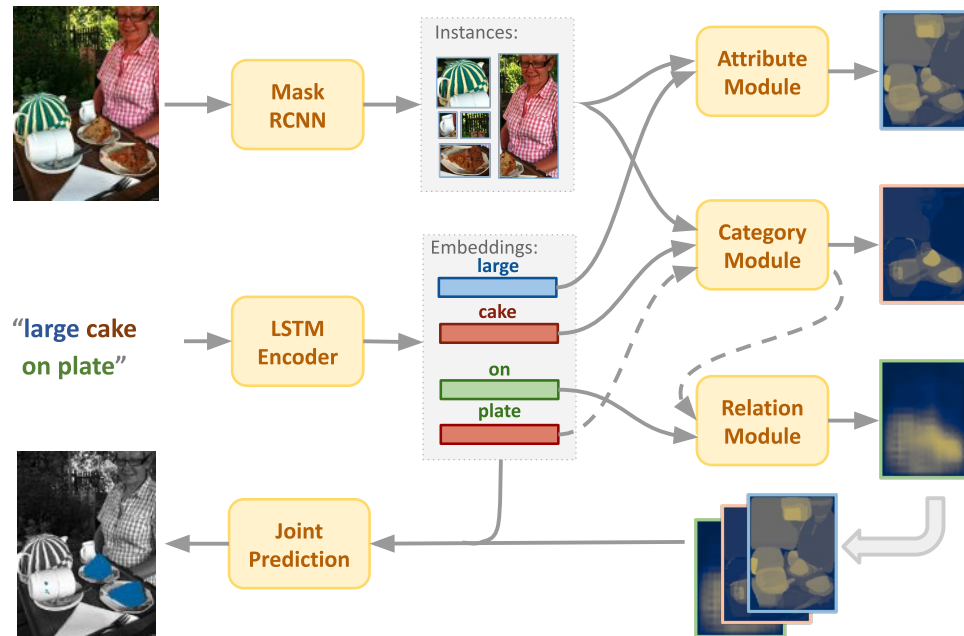


Figure 4.5: **Architecture of HULANet.** The architecture consists of modules to obtain attribute, category, and relation predictions given a phrase and an image. The attribute and category scores are obtained from Mask-RCNN detections and projected back to the image. The scores across categories and attributes are combined using a module-specific attention model. The relationship module is a convolutional network that takes as input the prediction mask of the related category and outputs a spatial mask given the relationship predicate. The modules are activated based on their presence in the query phrase and combined using an attention mechanism guided by the phrase.

Backbone encoders We use the Mask-RCNN [46] detector and bi-directional LSTMs [48] as our backbone encoders for images and phrases respectively. The Mask-RCNN (with ResNet101 [47] backbone) is trained to detect instances and predict category scores for the 1,272 categories that have a frequency over 20 on our dataset. Different from instance detection tasks on standard benchmarks, we allow relatively noisy instance detections by setting a low threshold on objectness scores and by allowing at most 100 detections per image to obtain a high recall. For phrase encoding, we train three separate bi-directional LSTMs to generate embeddings for categories, attributes and relationship phrases. They share the same word embeddings initialized from FastText [20] as the input to the LSTM, and have mean pooling applied on the LSTM output of the corresponding words as the encoded output.

Category module The category module takes as input the phrase embedding of the category and detected instance boxes (with masks) from Mask-RCNN, and outputs a score-map of corresponding regions in the image. We first construct the category channels $C \in \mathbb{R}^{N \times H \times W}$ by projecting the Mask-RCNN predictions back to the image. Here $N = 1272$ is the number of categories and $H \times W$ is set to $1/4 \times$ the input image size. Concretely, for each instance i detected by Mask R-CNN as category c_i with score s_i , we project its predicted segmentation mask to image as a binary mask $m_{i,H \times W}$, and update the category channel score at the corresponding location as $C[c_i, m_i] := \max(s_i, C[c_i, m_i])$. Finally, each category channel is passed through a “layer-norm” which scales the mean and variance of each channel.

To compute the attention over the category channels, the phrase embedding e_{cat} is passed through a few linear layers f with sigmoid activation at the end to predict the attention weights over the category channels $A = \sigma(f(e_{cat}))$. We calculate the weighted sum of the category channels guided by the attention weights $S_{H \times W} = \sum_c A_c \cdot C_c$, and apply a learned affine transformation plus sigmoid to obtain the category module prediction heatmap $P_{H \times W} = \sigma(a \cdot S_{H \times W} + b)$. This attention scheme enables the category module to

leverage predictions from good category detectors to improve performance on more difficult categories. We present other baselines for combining category scores in the ablation studies in Section 4.3.

Attribute module The attribute module is similar to the category module except for an extra attribute classifier. On top of the pooled ResNet instance features from Mask-RCNN, we train a two-layer multi-label attribute classifier. To account for significant label imbalance we weigh the positive instances more when training attribute classifiers with the binary cross-entropy loss. To obtain attribute score channels we take the top 100 detections and project their top 20 predicted attributes back to the image. Identical with the category module, we use the instance masks from the Mask-RCNN, update the corresponding channels with the predicted attribute scores, and finally apply the attention scheme guided by the attribute embedding from the phrase to obtain the final attribute prediction score heat-map.

Relationship module Our simple relationship module uses the category module to predict the locations of the supporting object. The down-scaled (32×32) score of the supporting object is concatenated with the embedding of the relationship predicate. This is followed by two dilated convolutional layers with kernel size 7 applied on top, achieving a large receptive field without requiring many parameters. Finally, we apply an affine transformation followed by sigmoid to obtain the relationship prediction scores. The convolutional network can model coarse spatial relationships by learning filters corresponding to each spatial relation. For example, by dilating the mask one can model the relationship “near”, and by moving the mask above one can model the relationship “on”.

Combining the modules The category, attribute, and relation scores P_c, P_a, P_r obtained from individual modules are each represented as a $H \times W$ image, $1/4$ the image size. To this we append channels of quadratic interactions $P_i \circ P_j$ for every pair of channels (including $i = j$), obtained using elementwise product and normalization, and a bias channel of all ones, to obtain a 10-channel scoremap F (3+6+1 channels). Phrase embeddings

of category, attribute and relationship are concatenated together and then encoded into 10-dimensional “attention” weights w through linear layers with LeakyReLU and DropOut followed by normalization. When there is no attribute or relationship in the input phrase, the corresponding attention weights are set to zero and the attention weights are re-normalized to sum up to one. The overall prediction is the attention-weighted sum of the linear and quadratic feature interactions: $O = \sum_t F_t w_t$. Our experiments show a slight improvement of 0.05% on validation `mean-IoU` with the quadratic features.

Training details The Mask-RCNN is initialized with weights pre-trained on the MS-COCO dataset [70] and fine-tuned on our dataset. It is then fixed for all the experiments. The attribute classifier is trained on ground-truth instances and their box features pooled from Mask-RCNN with a binary cross-entropy loss specially weighted according to attribute frequency. These are also fixed during the training of the referring modules. On top of the fixed Mask-RCNN and the attribute classifier, we separately train the individual category and attribute modules. When combining the modules we initialize the weights from individual ones and fine-tune the whole model end-to-end. We apply a pixel-wise binary cross-entropy loss on the prediction score heat-map from each module and also on the final prediction heat-map. To account for the evaluation metric (`mean-IoU`), we increase the weights on the positive pixels and average the loss over referring phrase-image pairs instead of over pixels. All our models are trained on the training set. For evaluation, we require a binary segmentation mask which is obtained by thresholding on prediction scores. These thresholds are set based on `mean-IoU` scores on the validation set. In the next section, we report results on the test set.

Model	mean-IoU	cum-IoU	Pr@0.5	Pr@0.7	Pr@0.9
HULANet					
cat	39.9	48.8	40.8	25.9	5.5
cat+att	41.3	50.8	42.9	27.8	5.9
cat+rel	41.1	49.9	42.3	26.6	5.6
cat+att+rel	41.3	50.2	42.4	27.0	5.7
Mask-RCNN self	36.2	45.9	37.2	22.9	4.1
Mask-RCNN top	39.4	47.4	40.9	25.8	4.8
RMI	21.1	42.5	22.0	11.6	1.5
MattNet	20.2	22.7	19.7	13.5	3.0

Table 4.1: **Comparison of various approaches on the entire test set of VGPHRASE-CUT.** We compare different combinations of modules in our approach (HULANet) against baseline approaches: Mask-RCNN, RMI and MattNet.

Model	all	coco	1-100	101-500	500+
HULANet					
cat	39.9	46.5	46.8	31.8	25.2
cat+att	41.3	48.3	48.2	33.6	26.6
cat+rel	41.1	47.9	47.8	33.6	26.6
cat+att+rel	41.3	47.8	47.8	33.8	27.1
Mask-RCNN self	36.2	44.9	45.5	27.9	10.1
Mask-RCNN top	39.4	46.1	46.4	31.6	23.2
RMI	21.1	23.7	28.4	12.7	5.5
MattNet	20.2	19.3	24.9	14.8	10.6

Table 4.2: **The mean-IoU on VGPHRASECUT test set for various category subsets.** The column *coco* refers to the subset of data corresponding to the 80 coco categories, while the remaining columns show the performance on the top 100, 101-500 and 500+ categories in the dataset sorted by frequency.

Model	all	att	att+	rel	rel+	stuff	obj
HULANet							
cat	39.9	37.6	37.4	32.3	33.0	47.2	33.9
cat+att	41.3	39.1	38.8	33.7	33.8	48.4	35.5
cat+rel	41.1	38.8	38.4	33.8	34.0	48.1	35.4
cat+att+rel	41.3	39.0	38.5	34.1	33.9	48.3	35.6
Mask-RCNN self	36.2	34.5	34.7	29.0	30.8	44.4	29.5
Mask-RCNN top	39.4	37.3	36.6	31.9	32.6	46.4	33.6
RMI	21.1	19.0	21.0	11.6	12.2	31.1	13.0
MattNet	20.2	19.0	18.9	15.6	15.1	25.5	16.0
Model	all	single	multi	many	small	mid	large
HULANet							
cat	39.9	41.2	37.0	34.3	15.1	40.3	67.6
cat+att	41.3	42.6	38.6	35.9	17.1	42.0	68.0
cat+rel	41.1	42.5	38.2	35.5	17.1	41.5	68.2
cat+att+rel	41.3	42.6	38.4	35.7	17.3	41.7	68.2
Mask-RCNN self	36.2	37.2	34.1	29.9	17.0	35.7	59.4
Mask-RCNN top	39.4	40.6	36.8	33.4	18.5	39.3	63.6
RMI	21.1	23.1	16.9	12.7	1.2	18.6	49.5
MattNet	20.2	22.2	15.9	12.6	6.1	18.9	39.5

Table 4.3: **The mean-IoU on VGPHRASECUT test set for additional subsets.** *att/rel*: the subset with attributes/relationship annotations; *att+/rel+*: the subset which requires attributes or relationships to distinguish the target from other instances of the same category; *single/multi/many*: subsets that contain different number of instances referred by a phrase; *small/mid/large*: subsets with different sizes of the target region.

4.3 Results and Analysis

4.3.1 Comparison to baselines

Table 4.1 shows the overall performance of our model and its ablated versions with two baselines: RMI [73] and MattNet [128]. They yield near state-of-the-art performance on datasets such as RefCOCO [60].

RMI is a single-stage visual grounding method. It extracts spatial image features through a convolutional encoder, introduces convolutional multi-modal LSTM for jointly modeling of visual and language clues in the bottleneck, and predicts the segmentation through an upsampling decoder. We use the RMI model with ResNet101 [47] as the image encoder. We initialized the ResNet with weights pre-trained on COCO [70], trained the whole RMI model on our training data of image region and referring phrase pairs following the default setting as in their public repository, and finally evaluated it on our test set.

RMI obtains high cum-IoU but low mean-IoU scores because it handles large targets well but fails on small ones (see Table 4.3 “small/mid/large” subsets). cum-IoU is dominated by large targets while our dataset many small targets: 20.2% of our data has the target region smaller than 2% of the image area, while the smallest target in RefCOCO is 2.4% of the image. Figure 4.6 also shows that RMI predicts empty masks on challenging phrases and small targets.

MattNet focuses on ranking the referred box among candidate boxes. Given a box and a phrase, it calculates the subject, location, and relationship matching scores with three separate modules, and predicts attention weights over the three modules based on the input phrase. Finally, the three scores are combined with weights to produce an overall matching score, and the box with the highest score is picked as the referred box.

We follow the training and evaluation setup described in their paper. We train the Mask-RCNN detector on our dataset, and also train MattNet to pick the target instance box among ground-truth instance boxes in the image. Note that MattNet training relies on complete annotations of object instances in an image, which are used not only as the candidate boxes

but also as the context for further reasoning. The objects in our dataset are only sparsely annotated, hence we leverage the Visual Genome boxes instead as context boxes. At test time the top 50 Mask-RCNN detections from all categories are used as input to the MattNet model.

While this setup works well on RefCOCO, it is problematic on VGPHRASECUT because detection is more challenging in the presence of thousands of object categories. MattNet is able to achieve $\text{mean-IOU} = 42.4\%$ when the ground-truth instance boxes are provided in evaluation, but its performance drops to $\text{mean-IOU} = 20.2\%$ when Mask-RCNN detections are provided instead. If we only input the detections of the referred category to MattNet, mean-IOU improves to 34.7%, approaching the performance of *Mask-RCNN self*, but it still performs poorly on rare categories.

Our modular approach for computing robust category scores from noisy detections alone (*HULANet cat*) outperforms both baselines by a significant margin. Example results using various approaches are shown in Figure 4.6.

4.3.2 Ablation studies and analysis

Table 4.2 shows that the performance is lower for rare categories. Detection of thousands of categories is challenging, but required to support open-vocabulary natural language descriptions. However, natural language is also redundant. In this section we explore if a category can leverage scores from related categories to improve performance, especially when it is rare.

First we evaluate Mask-RCNN as a detector, by using the mask of the top-1 detected instance from the referred category as the predicted region. The result is shown as the row “*Mask-RCNN self*” in Table 4.2. The row below “*Mask-RCNN top*” shows the performance of the model where each category is matched to a single other category based on the best mean-IOU on the training set. For example, a category “pedestrian” may be matched to “person” if the person detector is more reliable. As one can see in Table 4.2,

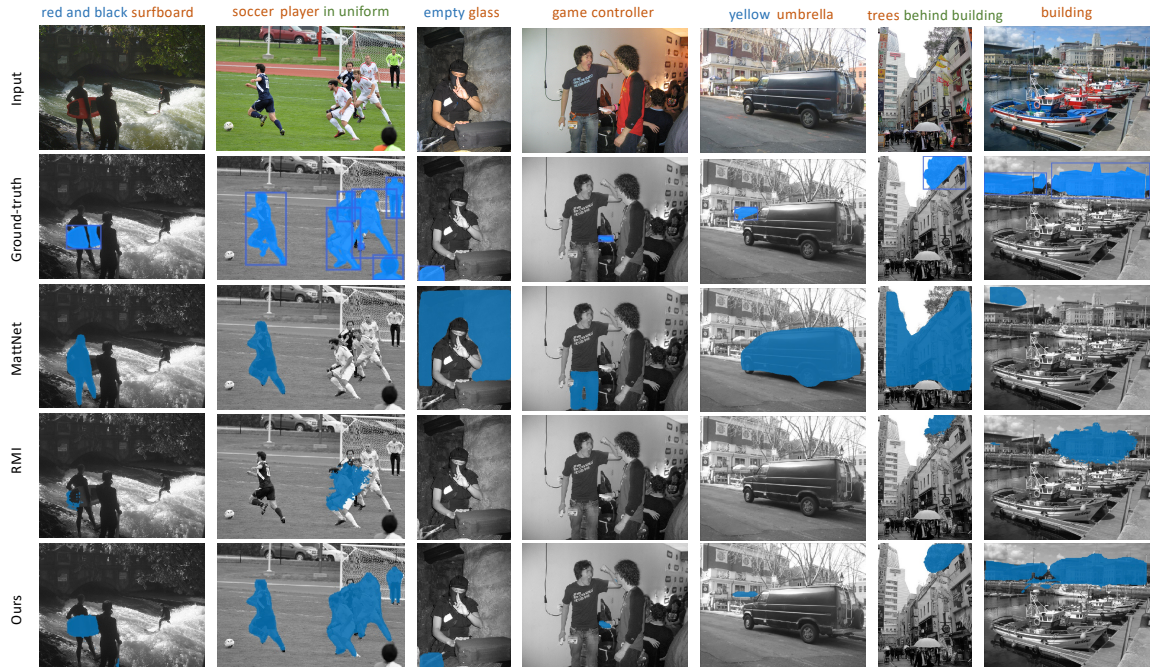


Figure 4.6: **Prediction results on VGPHRASECUT dataset.** Rows from top to down are: (1) input image; (2) ground-truth segmentation and instance boxes; (3) MattNet baseline; (4) RMI baseline; (5) HULANet (cat + att + rel).

the performance on the tail categories jumps significantly (10.1% \rightarrow 23.2% on the 500+ subset.) In general the tail category detectors are poor and rarely used. This also points to a curious phenomenon in referring expression tasks where even though the named category is specific, one can get away with a coarse category detector. For example, if different animal species never appear together in an image, one can get away with a generic animal detector to resolve any animal species.

This also explains the performance of the category module with the category-level attention mechanism. Compared to the single category picked by the Mask-RCNN top model, the ability of aggregating multiple category scores using the attention model provides further improvements for the tail categories. Although not included here, we find a similar phenomenon with attributes, where a small number of base attributes can support a larger, heavy-tailed distribution over the attribute phrases. It is reassuring that the number of visual concepts to be learned grows sub-linearly with the number of language

concepts. However, the problem is far from solved as the performance on tail categories is still significantly lower.

Table 4.3 shows the results on additional subsets of the test data. Some high-level observations are that: (i) Object categories are more difficult than stuff categories. (ii) Small objects are extremely difficult. (iii) Attributes and relationships provide consistent improvements across different subsets. Remarkably, the improvements from attributes and relationships are more significant on rare categories and small target regions where the category module is less accurate.

4.3.3 Modular heatmap visualization

In Figure 4.7 and Figure 4.8, we show HULANet predictions and modular heatmaps.

Figure 4.7 demonstrates that our attribute module is able to capture color (“black”, “brown”), state (“closed”), material (“metal”) and long and rare attributes (“pink and white”). In the first (“black jacket”) example, the category module detects two jackets, while the attribute module is able to select out the “black” one against the white one.

Figure 4.8 shows how our relationship module modifies the heatmaps of supporting objects depending on different relationship predicates. With the predicate “wearing”, the relationship module predicts expanded regions of the detected “jacket” especially vertically. The relational prediction of “parked on” includes regions of the “street” itself as well as regions directly above the “street”, while the predicate “on” leads to the identical region prediction as the supporting object. In the last example of “sitting at”, a broader region around the detected “table” is predicted, covering almost the whole image area.

4.3.4 Failure case analysis

Figure 4.9 displays typical failure cases from our proposed HULANet. Heatmaps from internal modules provide more insights where and why the model fails.

In the first example, our backbone Mask-RCNN fails to detect the ground-truth “traffic cones”, which are extremely small and from rare categories. Similarly, in the second

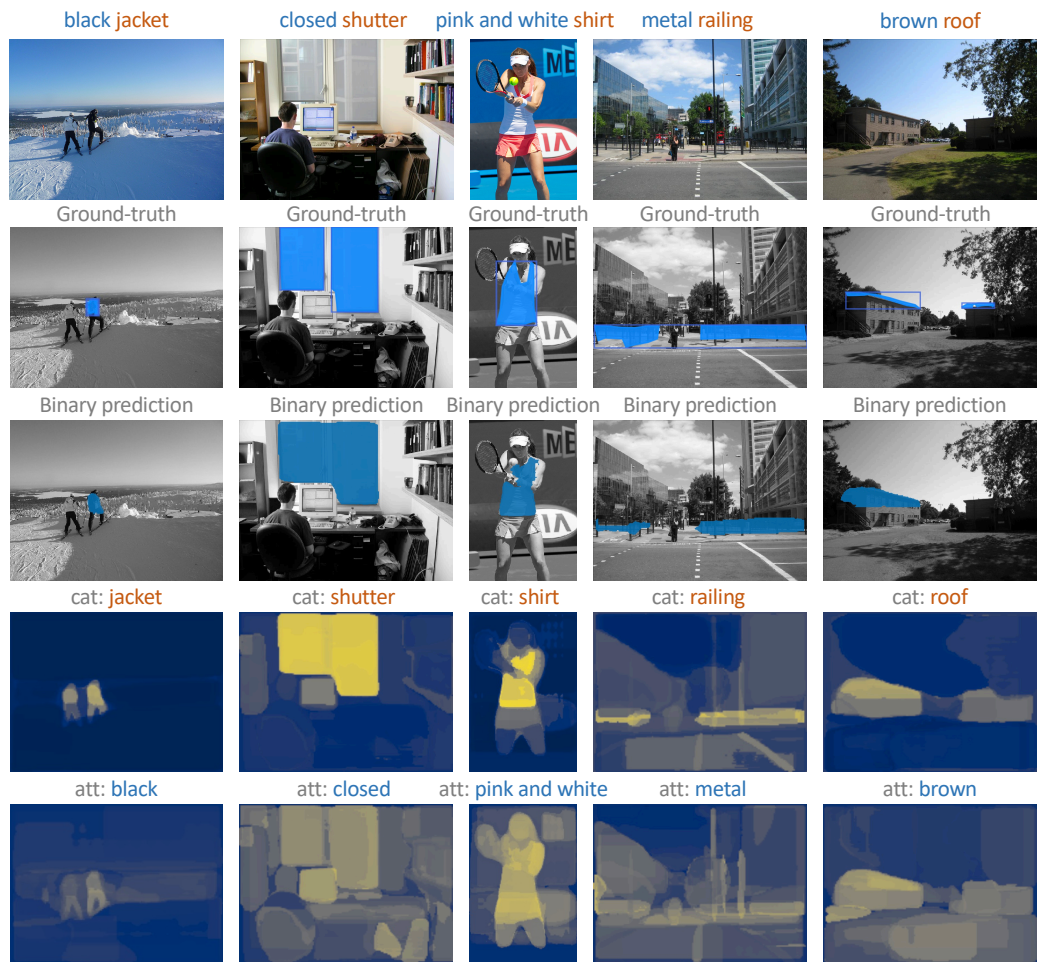


Figure 4.7: **HULANet prediction results and heatmaps on phrases with attributes.** Rows from top to down are: (1) input image; (2) ground-truth segmentation and instance boxes; (3) predicted binary mask from HULANet (cat+att+rel); (4) heatmap prediction from the category module; (5) heatmap prediction from the attribute module.

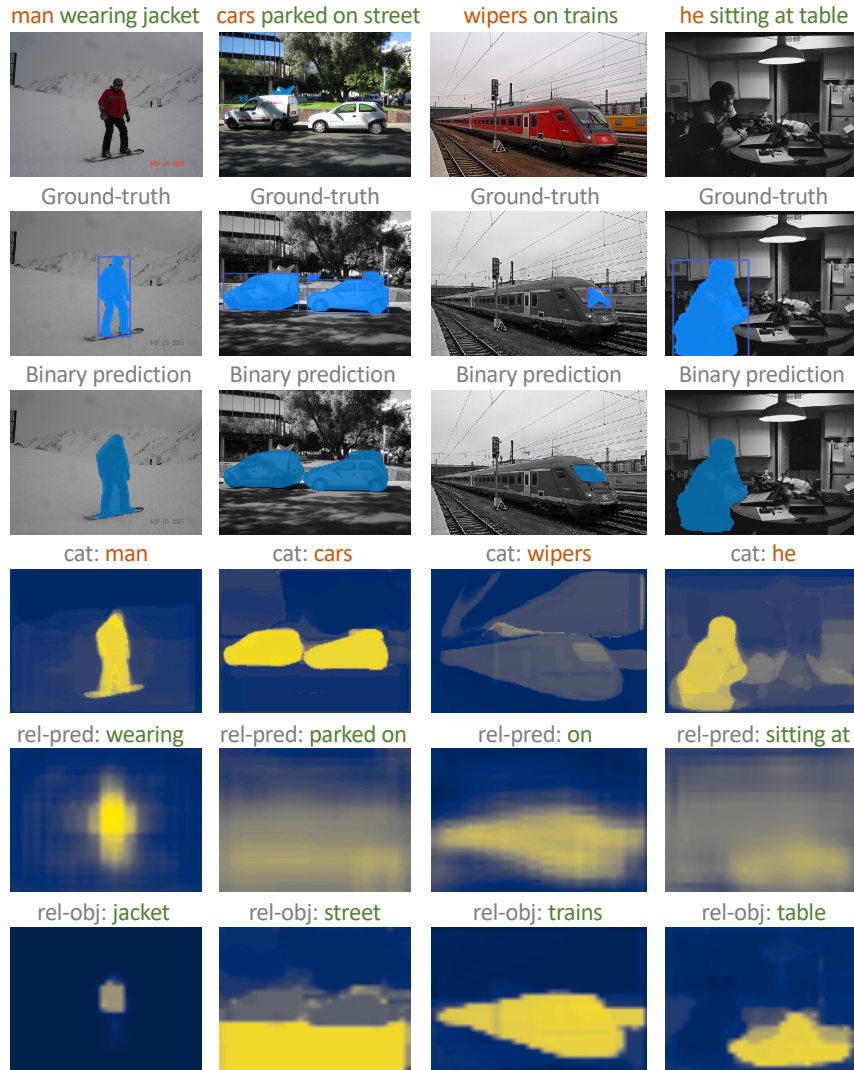


Figure 4.8: **HULANet prediction results and heatmaps on phrases with relationships.** Rows from top to down are: (1) input image; (2) ground-truth segmentation and instance boxes; (3) predicted binary mask from HULANet (cat+att+rel); (4) heatmap prediction from the category module; (5) heatmap prediction from the relationship module; (6) heatmap prediction of the supporting object (in the relationship description) from the category module.

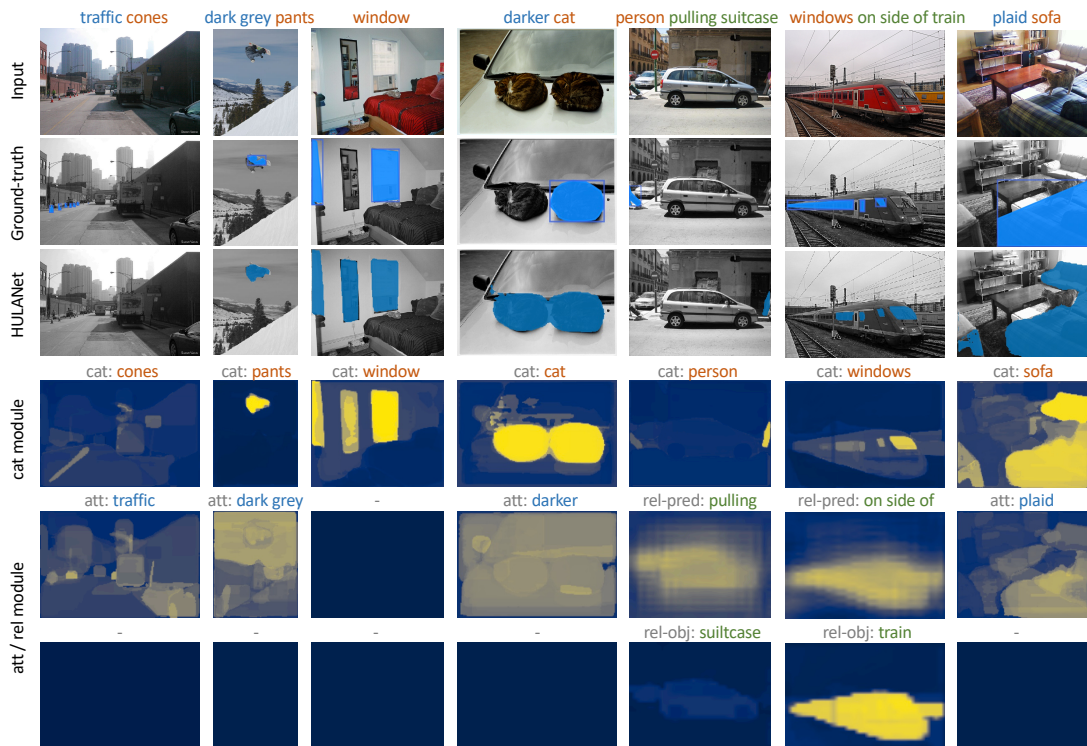


Figure 4.9: **Negative results from HULANet on VGPHRASECUT test set.** Rows from top to down are: (1) input image; (2) ground-truth segmentation and instance boxes; (3) predicted binary mask from HULANet (cat+att+rel); (4) heatmap prediction from the category module; (5-6) heatmap predictions from additional (attribute or relationship) modules.

“dark grey pants” example, the “pants” is not detected as a separate instance in the backbone Mask-RCNN, therefore the category module can only predict the whole mask of the skateboarder.

The third “window” example shows when the category module (and the backbone Mask-RCNN) fails to distinguish mirrors from windows. In the fourth example, our attribute module fails to recognize which cat is “darker” than the other.

We then display two failure cases for the relationship module. It fails on the first one because the supporting object (“suitcase”) is not detected by the category module, and fails on the second one for unable to accurately model the relation predicate “on side of”.

In the last example, although our attribute module figures out which sofa is “plaid”, the final prediction is dominated by the category module and fails to exclude non-plaid sofas.

4.4 Summary

We presented a new dataset, VGPHRASECUT, to study the problem of grounding natural language phrases to image regions. By scaling the number of categories, attributes, and relations we found that existing approaches that rely on high-quality object detection show a dramatic reduction in performance. Our proposed HULANet performs significantly better, suggesting that dealing with long-tail object categories via modeling their relationship to other categories, attributes, and spatial relations is a promising direction of research. Another take away is that decoupling representation learning and modeling long-tails might allow us to scale object detectors to rare categories, without requiring significant amount of labelled visual data. Nevertheless, the performance of the proposed approach is still significantly below human performance which should encourage better modeling of language and vision.

CHAPTER 5

EVALUATING LARGE-SCALE LANGUAGE-VISION MODELS

There has been significant progress on training large-scale models such as ResNet [47] on ImageNet [34] for vision, and BERT [35], GPT-3 [26] trained on WebText [95] for natural language understanding. However, large-scale models that jointly understand multiple modalities, such as language and vision, have been lacking in comparison. Therefore the common strategy for language and vision tasks, including the ones we used in prior chapters, was to align pre-trained models for each modality using domain-specific aligned data. This allows the benefit of transfer learning on each modality but requires collecting training data and fine-tuning for each cross-modal task.

Recently, this has changed with the publication of models that can have a common understanding of language and vision such as CLIP [94]. CLIP is a model trained on a massive dataset of images paired with text that learns to embed language and vision data into a common embedding space. It has been applied to downstream tasks such as geo-localization, optical character recognition, facial emotion recognition, and action recognition as introduced in [94].

We investigate how well the CLIP representations generalize to novel vision-language tasks especially in fine-grained domains such as those we have considered in Chapter 2 and 3. While the analysis in [94] is more focused on image categorization based on the category names alone, we look further into CLIP’s capability of understanding adjectives or attribute phrases in fine-grained domains. We expect to see similar benefits of transfer learning in these cross-modal tasks as one has observed in language and vision tasks individually.

Specifically, we analyze the capability of CLIP on:

1. Recognizing fine-grained differences between two images. As shown in Figure 2.1, given two phrases “P1 vs. P2” describing the difference between two images, the goal is to figure out which image is described by “P1” and the other image by “P2”.
2. Image and phrase retrieval in specific domains. Given an attribute phrase, the task is to retrieve images from a domain-specific dataset that match with the input phrase, and vice versa for phrase retrieval.
3. Understanding how well CLIP handles compositionality of natural language. We analyze the image retrieval performance with compositional phrase queries describing the combination of two colors, color plus pattern, as well as foreground/background colors on the *synthetic texture dataset* introduced in Chapter 3 Section 3.3.3
4. Leveraging attributes to improve fine-grained classification accuracy in a zero-shot setting. While CLIP was demonstrated to be able to construct zero-shot classifiers based on the name of the class alone, we investigate if the image and class level describable attributes can be incorporated to boost the performance further.

5.1 Models and Datasets

We compare CLIP with models with the best performance from Chapter 2 and 3. Each of the three models include an image encoder and a text encoder to encode images/texts into a shared embedding space, and provides a distance/similarity function such that relative/paired images and texts have smaller distances in the embedding space than irrelevant images and texts.

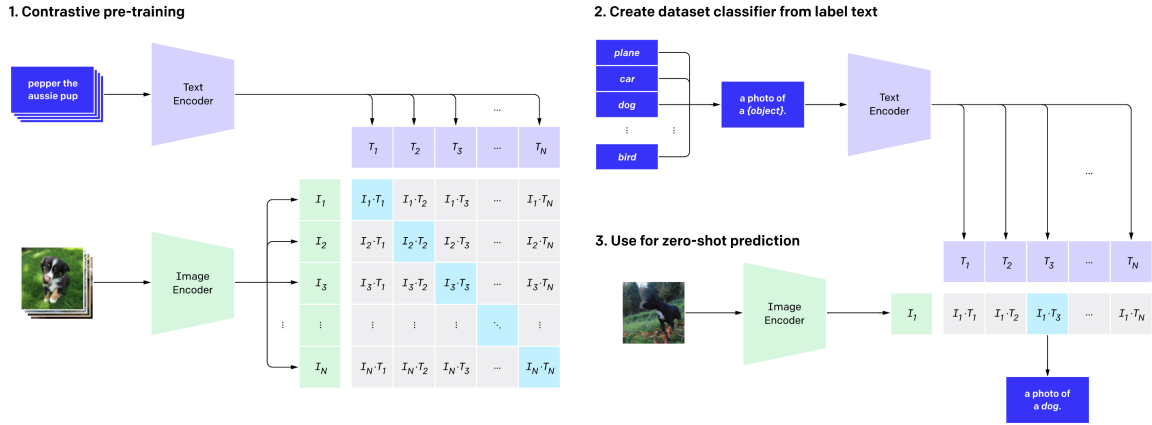


Figure 5.1: **Summary of CLIP model.** Figure is from [94] Figure 1: “CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset’s classes.”

5.1.1 Overview of CLIP

CLIP is introduced in [94]. As shown in Figure 5.1, an image encoder and a text encoder are jointly trained on 400 million image-text pairs collected from the Internet ¹ Given an image, the training objective task is to score the image and text pairs to be higher than other text and other images in the sampled batch. They experimented with various image and text encoding architectures to concluded that Transformer [111] text encoder and vision Transformer [39] as the image encoder yield the best performance. Our experiments in this Chapter is on the model with Transformer text encoder and “ViT-B/32” vision transformer.

CLIP can be used as a zero-shot classifier as introduced in [94]. In zero-shot learning the goal is to learn an image classifier given a description of the categories. For example, each category can be encoded by the name and the text encoder of CLIP can map an input in the format of “A photo of a [Category]” to an embedding that is compatible with the image embeddings. For a given image, one can therefore search for its nearest category embedding to predict its predicted category. The authors experimented on 27 datasets to

¹The dataset is not yet publicly available

demonstrate that CLIP is competitive with fully supervised baselines, suggesting that CLIP has a strong understanding of good understanding of common categories. In Section 5.2.5, we investigate if the performance can be further improved using detailed description of categories.

5.1.2 Baselines

From Chapter 2 we choose the “Simple Listener” with LSTM [102] text encoder and VGG-16 [105] image encoder (noted as “OID-SL”). The text encoder is trained from scratch. The VGG-16 image encoder is pre-trained on ImageNet [66] and then fine-tuned on the OID Attribute Phrases Dataset.

From Chapter 3 we choose the metric learning model with BERT [35] text encoder and ResNet101 [47] image encoder (noted as “DTD2-ML”). The BERT model is pre-trained on “BookCorpus” [134] and “English Wikipedia” and does not get updated when training on DTD². Only a linear layer on top of BERT is trained on DTD² Dataset. The ResNet-101 model is pre-trained on ImageNet and fine-tuned on DTD².

Table 5.1 compares the size of the selected models and their training data sets. CLIP has a comparable size of training data to the size of ImageNet and BERT but it sees much more paired data than either OID-SL or DTD2-ML has seen. It allows CLIP to train more complicated encoding models from scratch.

In addition to the domains of aircraft and textures, we also conduct analysis on the Caltech-UCSD Birds (CUB) [116] which contains 11,788 images across 200 bird species with 312 binary attributes on each image.

5.2 Experiments and Analysis

5.2.1 Recognizing fine-grained differences between aircraft images

CLIP can be used to discriminate between a pair of images given a natural language description (i.e., a listener model as described in OID Attribute Phrases Dataset as introduced

	Model	CLIP	OID-SL	DTD2-ML
Training Set Size	Vocabulary	49152	730*	1673*
	Images	$\leq 400\text{M}$	1851	3222
	Text-img pairs	400M	4700	14797
Trainable parameters	Image encoder	86M	138M**	45M**
	Text encoder	63M	4M	0.2M

*Only words with frequency of at least 5 are counted.

** Fine-tuning on top of ImageNet pre-trained models.

Table 5.1: **Comparison of model and training data size.** Note that the image encoders of OID-SL and DTD2-ML are pre-trained on ImageNet with 14M images. The text encoder of DTD2-ML contains BERT with 110M parameters which is not updated during training, and we only train a linear layer on top of it.

in Chapter 2.) Given two images (I_1 and I_2) and text input in the format of “Phrase 1 (P_1) vs. Phrase 2 (P_2)”, the task is to figure out which image between I_1 and I_2 is described by P_1 (and the other image is described by P_2).

We construct the text input from templates such as “An image of an aircraft with [P]” where [P] is the phrase of interest. We compute the cosine similarity $S(I, P)$ between the embedding of image I and the embedding of the sentence constructed from phrase P :

$$S(I, P) = \frac{\phi(I) \cdot \theta(P^*)}{\|\phi(I)\|_2 \|\theta(P^*)\|_2}$$

Where $\phi(\cdot)$ is the image encoder, $\theta(\cdot)$ is the text encoder, and P^* is the sentence constructed from P using the aforementioned template. With input images I_1, I_2 and phrases P_1, P_2 , we predict “ I_1 is described by P_1 , I_2 is described by P_2 ” if:

$$S(I_1, P_1) + S(I_2, P_2) - S(I_2, P_1) - S(I_1, P_2) > 0$$

and predict “ I_2 is described by P_1 , I_1 is described by P_2 ” otherwise.

The effect of the choice of the template on the task performance in the validation set is shown in Table 5.2. The best template was found to be “An image of an aircraft with [Phrase]”. CLIP reaches an accuracy of 72.43% on the test set of OID Attribute Phrases.

Template	Accuracy
[Phrase]	71.87
An [Phrase] airplane	71.74
An airplane with [Phrase]	72.17
An airplane that is [Phrase]	72.12
An airplane that has [Phrase]	72.11
An image of an [Phrase] airplane	73.10
An image of an airplane with [Phrase]	73.31
An image of an airplane that is [Phrase]	73.21
An image of an airplane that has [Phrase]	73.12
A photo of an airplane with [Phrase]	72.86
An image of a plane with [Phrase]	73.44
An image of an aircraft with [Phrase]	73.45
An image of a flight with [Phrase]	72.12
An image of a jet with [Phrase]	72.52

Table 5.2: **Referring accuracy of CLIP on OID Attribute Phrases validation set with different input text templates.** According to the text templates we have evaluated, the templates have a small impact of 1.7% on referring accuracy. “An image of an aircraft with [Phrase]” is the best template we have found.

The Simple Listener trained on OID Attribute Phrases has an accuracy 89.3% for a comparison (See Table 2.1). This is remarkable as the CLIP models have not been fine-tuned with the domain-specific data.

Figure 5.2 shows the easiest and most difficult phrases for CLIP based on the referring accuracy when having each phrase in the input. Figure 5.3 displays failure cases for CLIP. While CLIP is good at understanding colors (“military plane” is mostly gray in color) and the scene (e.g., “on tarmac”, “on grass”). It has a poorer understanding of parts, their relations, and pose (e.g., “open cockpit”, “fewer windows”) and spatial/location descriptions (“facing right”, “wings on top”).

5.2.2 Phrase and Image Retrieval on DTD²

We apply CLIP to phrase and image retrieval on DTD² the same as in Chapter 3 Section 5.1. CLIP takes input in the format of “An image of [P] texture”, where [P] is the input attribute phrase. The template is selected based on the retrieval performance on DTD²

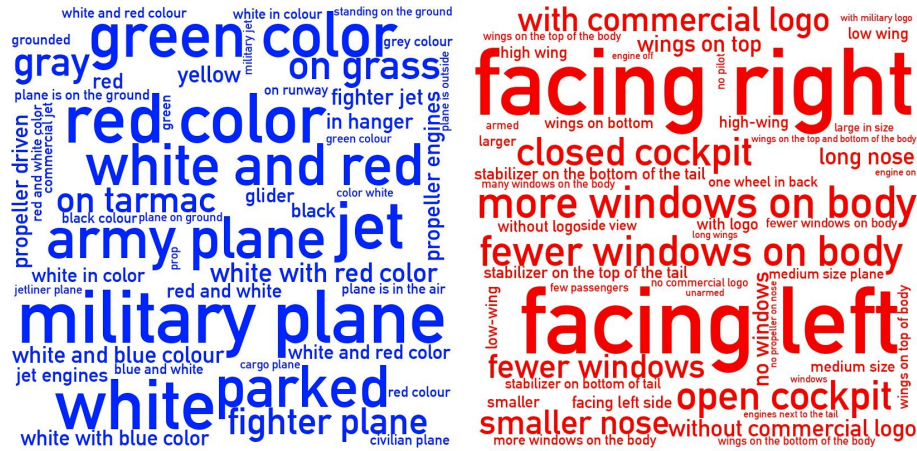


Figure 5.2: **Phrases with best and worst referring accuracy for CLIP on OID Attribute Phrases referring task.** For each phrase P , we calculate the referring accuracy on data when P is one of the two input phrases. Left: 50 phrases with highest accuracy ($\geq 95\%$); Right: 50 phrases with lowest accuracy ($\leq 51\%$). Font size is proportional to the phrase frequency in OID Attribute Phrases Dataset.

validation set as shown in Table 5.3. It outperforms directly using the phrase by a large margin.

We use CLIP encoders to calculate embeddings for every image in DTD² test set and text inputs constructed from each of the 655 attribute phrases from DTD². For image retrieval, we search for the nearest images from each input phrase; For phrase retrieval, we search for nearest phrases from each input image embedding.

In Table 5.4 we compare the retrieval metrics between CLIP and DTD2-ML. Compared with DTD2-ML, CLIP gets similar performance on image retrieval but performs worse on phrase retrieval. In Figure 5.4 we display image retrieval examples. Both models retrieve reasonable images. “Zigzagged” is a very specific pattern that CLIP makes mistakes on but DTD2-ML is able to understand accurately. “Equally spaced” is a failure case for CLIP that it looks for galaxy space.

In Figure 5.5 we show phrase retrieval examples. Although the retrieval metrics for CLIP are low in Table 5.4, its retrieved phrases look reasonable. In the first example, both CLIP and DTD2-ML retrieve “striped”. DTD2-ML also retrieves “lined” and “lines”,



Figure 5.3: **Failure cases of CLIP on OID Attribute Phrases referring task.** For each example, the ground-truth in the Attribute Phrases Dataset indicates that the first phrase (before “vs/”) describes the image on the left, but CLIP predicts the opposite.

Template	Image Retrieval	Phrase Retrieval
[Phrase]	13.53	9.34
A [Phrase] image	13.54	9.93
[Phrase] texture	14.15	11.57
A photo of [Phrase] texture	13.82	11.98
An image of [Phrase] texture	13.93	12.31
An image with [Phrase] texture	12.88	10.65

Table 5.3: **Image and phrase retrieval mean average precision of CLIP on DTD² validation split.** Considering both image and phrase retrieval performance, we select “An image of [Phrase] texture” as our template for further experiments.

while CLIP retrieves “striated” and “strips”, which are all synonyms to “striped”. However, “striated” and “strips” are rare in DTD² and DTD2-ML learns from the statistic bias in DTD² vocabulary to not predict “striated” and “strips” but to predict more frequent words such as “lined” and “lines”. It’s a common issue for vision and language datasets that we can only collect partial annotations with a lot of attributes and descriptions that are actually true to a given image but not collected/labeled as correct. Since CLIP cannot leverage the dataset statistic bias, it is more often for CLIP to predict attributes that are reasonable but considered incorrect during evaluation. This explains the performance gap between CLIP and DTD2-ML on phrase retrieval.



Figure 5.4: **Image retrieval examples on DTD² test set from CLIP and DTD2-ML.** For each query attribute phrase, we display 5 random ground-truth images labeled with the given attribute, and the top 5 retrieved images from CLIP and DTD2-ML.

Figure 5.6 shows the phrases that each model is best or worst at. We calculate the image retrieval average precision for each phrase, plot the 80 phrases with the highest average precision as “positive” and the worst 80 phrases as “negative”. We also visualize phrases with the largest difference of average precision between CLIP and DTD2-ML. The two models share very similar easiest phrases that describe colors and most obvious textures, but their negative phrases are quite different. CLIP is better than DTD2-ML on colors such as “orange”, “pink”, “purple” that are basic but less frequent in DTD². CLIP also works better on attributes related to materials or certain types of objects (e.g., “wood”, “marble”, “glass”). However, CLIP performs worse on vocabulary specifically used to describe patterns and textures (e.g., “rough”, “lined”, “grooved”).

5.2.3 Phrase and Image retrieval of texture phrases on CUB Dataset

We further compare CLIP and DTD2-ML on CUB[116] Dataset which DTD2-ML hasn’t been trained on. We selected 17 attributes that both occur in DTD² and CUB as listed in 5.8. For example, images from CUB with attributes “has wing color: blue”, “has



Ground-truth phrases:

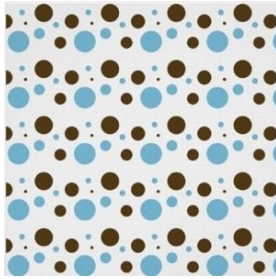
white, black, soft, smooth, grey, lined, striped, lines, black and white, horizontal, furry, large, animal print, fur

CLIP Retrieved top 20 (5 in ground-truth):

zebra, zebra print, **animal print**, stripe, **striped**, animal, stripes, striated, white and black, strips, **fur**, **black and white**, black & white color, barred, scratched, streaked, **furry**, silky, stripped, fuzzy

DTD2-ML Retrieved top 20 (10 in ground-truth):

black, **white**, **black and white**, white and black, **striped**, **lined**, **soft**, black & white color, **animal print**, **furry**, zebra, **lines**, banded, soft texture, **smooth**, zebra print, fuzzy, opaque, white background, grey



Ground-truth phrases:

white, black, brown, blue, spotted, dotted, circles, polka dotted, spots, on a cloth

CLIP Retrieved top 20 (5 in ground-truth):

polka dots, **spots**, white dots, **polka dotted**, dots, **spotted**, polka-dotted, spotty, bubbles, holes, **circles**, repeating pattern, **dotted**, pattern, geometric pattern, unpatterned, same design pattern, patterned surface, droplets, bubbly

DTD2-ML Retrieved top 20 (6 in ground-truth):

white, **circles**, round, circular, **dotted**, white background, small, **spotted**, polka-dotted, perforated, **black**, circular shape, multiple numbers, **polka dotted**, holey, dots, smooth, circle, holes, equally spaced

Figure 5.5: **Phrase retrieval examples on DTD² test set from CLIP and DTD2-ML.** For each image, we display its ground-truth phrases labeled in DTD² and top 20 retrieved phrases from CLIP and DTD2-ML. The bolded retrieved phrases are the ones included in the ground truth.

upperparts color: blue”, “has back color: blue”, etc. are all counted as positive samples for “blue”.

In Table 5.5 we show retrieval metrics of two models. CLIP performs better than DTD2-ML on image retrieval and they have similar performance on phrase retrieval. In Figure 5.7 we show image retrieval examples. CLIP is able to focus on the main object while DTD2-ML recognizes attributes from the background. For example, CLIP can retrieve “blue” birds, while DTD2-ML retrieved images with a “blue” background. On the other hand, DTD2-ML can retrieve from different categories but CLIP tends to return images of the same category, which implies that CLIP image features are highly related to categories such that images of the same category are close to each other in the embedding space. In Figure 5.8 we compare on each attribute in terms of image retrieval average precision. CLIP outperforms DTD2-ML on almost all attributes, especially “blue”, “yellow” and “red”.

Task	Model	MAP	MRR	P@5	P@20	R@5	R@20
Phrase Retrieval	DTD2-ML	31.68	72.59	40.67	22.96	20.23	44.50
	CLIP	12.06	31.41	15.82	12.63	5.63	16.28
Image Retrieval	DTD2-ML	13.50	31.12	16.52	14.57	5.24	17.32
	CLIP	12.21	39.96	17.73	11.44	8.45	21.61

Table 5.4: Compare the phrase retrieval and image retrieval performance of DTD2-ML and CLIP on DTD² test set.

Task	Model	MAP	MRR	P@5	P@20	R@5	R@20
Phrase Retrieval	DTD2-ML	52.58	68.65	46.36	-	45.80	-
	CLIP	53.36	74.54	42.40	-	42.46	-
Image Retrieval	DTD2-ML	35.33	53.71	44.71	43.82	0.17	0.75
	CLIP	50.10	94.12	74.12	71.76	0.48	1.57

Table 5.5: Compare the performance of phrase retrieval and image retrieval with DTD2-ML and CLIP on CUB test set. We experiment with 17 attributes that are included in both CUB and DTD² as input queries.

It is challenging to apply DTD2-ML to a novel domain of bird images because there is very limited overlapping of attributes and “bird” is an unseen concept for DTD2-ML. CLIP can perform reasonably on both DTD² and CUB datasets without any extra training. This demonstrates the strength of CLIP in generalizing to novel domains.

5.2.4 Understanding compositional phrases on synthetic texture images

We conduct the compositionality modeling analysis on synthetic texture images as described in Chapter 3 Section 5.3. Results are shown in Table 5.6.

We see a slight improvement for CLIP on “Background” compared against DTD2-ML but it performs lower than random guess on “Foreground”. Our interpretation is that background usually takes more area than the foreground. CLIP tends to recognize more of the majority color and fails to distinguish the foreground and background.

CLIP also achieves a slight improvement on “Color+Pattern” and a huge improvement on “Two-colors”. CLIP is trained on much more language data, therefore during training it may have seen plenty of examples of the combinations that are rare or novel in the DTD²

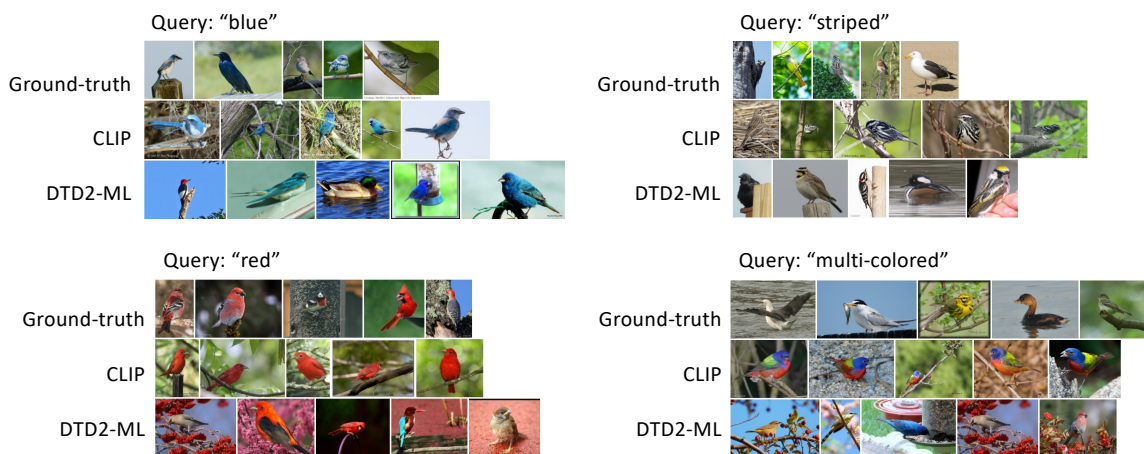


Figure 5.7: **Image retrieval examples on CUB test set from CLIP and DTD2-ML.** For each query attribute phrase, we display 5 random ground-truth images labeled with the given attribute, and the top 5 retrieved images from CLIP and DTD2-ML.

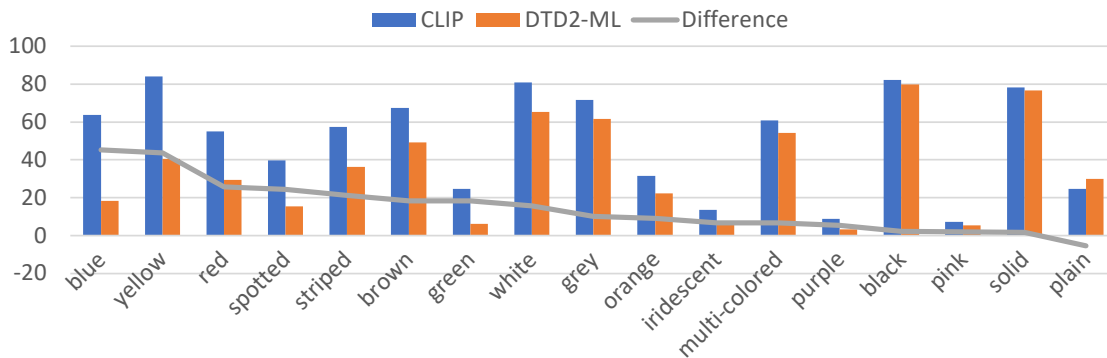


Figure 5.8: **Image retrieval average precision of each query attribute on CUB test set from CLIP and DTD2-ML.** The gray line shows the accuracy difference between CLIP and DTD2-ML. CLIP outperforms DTD2-ML on all attributes except “plain”.

Model	Foreground	Background	Color+Pattern	Two-colors
DTD2-ML	46.55±20.65	52.00±6.32	41.73±22.77	27.45±15.13
CLIP	38.00±14.94	60.18±5.49	45.23±23.51	55.18±16.18
Random guess	50.00	50.00	7.40	5.26

Table 5.6: Compare the R-Precision of image retrieval on texture compositional tasks with DTD2-ML and CLIP.

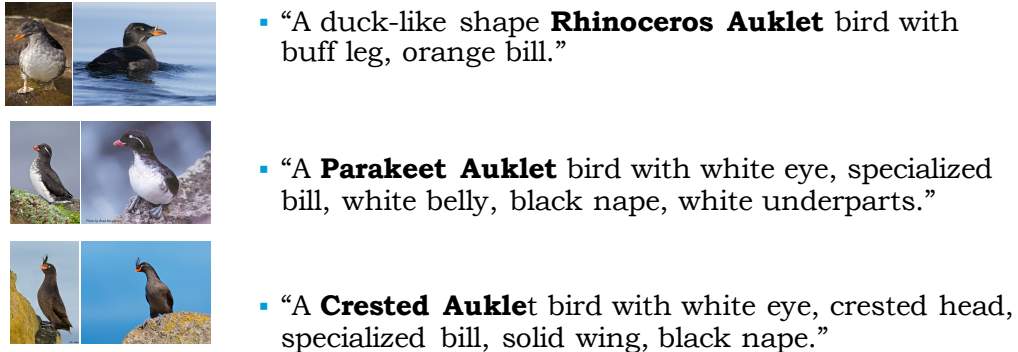


Figure 5.9: Examples of constructed category descriptions containing attributes. The category names are in bold. Although The three categories have similar names, the attributes can reflect their subtle differences between species, *e.g.*, “Rhinoceros Auklet” is more “duck-like” with “buff leg”, “Parakeet Auklet” has “white eye” and “white belly”, “Crested Auklet” has “crested head” and “black nape”.

dataset. While the DTD2-ML model needs to interpret the language composition, CLIP can simply learn the exact bi-gram or tri-gram phrases from training.

5.2.5 Zero-Shot Classification with Attribute Phrases

As introduced in [94], CLIP can work as a zero-shot classifier on novel categories. Each category C is converted to a sentence “A photo of a $[C]$ ” and encoded by the text encoder. Given an image, one can encode it with the image encoder and search for the nearest category as its prediction.

One concern of such a zero-shot classification mechanism is that the category names can be special proper nouns or rare words especially in fine-grained domains, which may lead to compromised classification performance. We have demonstrated the effectiveness

of attribute phrases for fine-grained classification in both Chapter 2 and 3. Here we extend the idea to improve fine-grained classification in the introduced zero-shot setting through leveraging attribute phrases to construct more informative category descriptions.

On CUB Dataset, we first count the percentage of images within a given category C in CUB that are positive for attribute A and compare it with the positive percentage of A across all categories. Based on such statistics, we gather a sorted list of attributes for each category that distinguishes it from other categories and add these attributes into the category description. The attributes from CUB contain adjectives describing a part of the bird, e.g. “has wing color::brown”, and we construct the category description in the format of “A [P_0] [C] (bird) with [P_1] [N_1], [P_2] [N_2], ...” where C is the category name, P_0 contains adjectives describing the whole bird, N_i are nouns of bird body parts and P_i are adjectives modifying N_i . The word “bird” is added only when the category name does not end with “bird”.

Figure 5.9 shows examples of generated descriptions with very similar category names. By using our constructed descriptions instead of only the category name (“A [C] (bird)”), we improve the classification accuracy slightly from 50.53% to **51.28%**.

We conduct the same experiment to classify 45 texture types in DTD². We count the most frequent attributes for each category in the training set, choose the top 20 attributes for each category that have a higher frequency than the average overall categories, and construct the phrases as “An image of [xxx] texture”, where [xxx] are the 20 phrases. For example, the “gauzy” class is described as “An image of gauzy, sheer, transparent, light, thin, white, translucent, soft, see through, delicate, netted, meshy, airy, silky, fabric, see-through, folded, wavy, curtains, cloth texture.” On DTD² test set, we achieve a classification accuracy of **54.84%**, compared to 41.06% when only including the category names.

The above experiments verify the effectiveness of CLIP for zero-shot classification as claimed in their paper. One can add a novel category to the CLIP classifier and achieve rea-

sonable performance with only a description of the category name and most distinguishing attributes, which is much easier to collect than a set of images for this novel category.

5.3 Summary

In this chapter, we analyze CLIP, a contrastive learning model with vision and language encoders trained on 400 million image-text pairs from the Internet, on specialized domains including aircraft, textures, and birds. Without any fine-tuning, CLIP achieves good performance on a wide range of language-vision tasks including image retrieval, text retrieval, and zero-shot classification. A detailed analysis of CLIP shows that the model is good at understanding coarse concepts, such as color and category names, but has worse performance on understanding fine-grained attributes such as parts, their relations, and pose. However, this can be alleviated by fine-tuning on domain-specific data.

CHAPTER 6

CONCLUSION

In this thesis, we leverage large-scale and detailed supervision from natural language to improve the understanding and modeling of visual domains. In Chapter 2, we propose to use *attribute phrases* to describe fine-grained visual differences between instances and learn to describe and ground these phrases to images in the context of a reference game. In Chapter 3, we focus on natural language that describes textures and address the challenge of capturing compositional properties (e.g., the combination of “color” and “pattern”). We train interpretable models on our proposed dataset and provide language-based explanations of texture features that are discriminative in fine-grained classification. In Chapter 5 we look into CLIP and demonstrate that a large-scale pre-trained model can achieve competitive performance in specialized domains. In the above three lines of work, we show that attribute phrases can capture detailed features and improve fine-grained classification. Lastly, in Chapter 4, we segment image regions based on referring phrases containing category names, attributes, and relationship descriptions between instances. We learn to model the associations between concepts to improve the handling of long-tail concepts. We introduce three datasets in Chapter 2, 3 and 4 which are all publicly available for the community.

BIBLIOGRAPHY

- [1] FastText pretrained embeddings <https://dl.fbaipublicfiles.com/fasttext/vectors-english/wiki-news-300d-1M.vec.zip>.
- [2] FGVC Butterflies and Moths Dataset, <https://sites.google.com/view/fgvc6/competitions/butterflies-moths-2019>.
- [3] Pretrained BERT of version “bert-base-uncased” https://huggingface.co/transformers/pretrained_models.html.
- [4] Pretrained ELMo https://allennlp.s3.amazonaws.com/models/elmo/2x4096_512_2048cnn_2xhighway/elmo_2x4096_512_2048cnn_2xhighway_weights.hdf5.
- [5] Abadi, Martin, Agarwal, Ashish, Barham, Paul, Brevdo, Eugene, Chen, Zhifeng, Citro, Craig, Corrado, Greg S, Davis, Andy, Dean, Jeffrey, Devin, Matthieu, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).
- [6] Akata, Zeynep, Reed, Scott, Walter, Daniel, Lee, Honglak, and Schiele, Bernt. Evaluation of output embeddings for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).
- [7] Akbari, Hassan, Karaman, Svebor, Bhargava, Surabhi, Chen, Brian, Vondrick, Carl, and Chang, Shih-Fu. Multi-level multimodal common semantic space for image-phrase grounding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
- [8] Amadasun, Moses, and King, Robert. Textural features corresponding to textural properties. *IEEE Transactions on systems, man, and Cybernetics* 19, 5 (1989), 1264–1274.
- [9] Anderson, Peter, He, Xiaodong, Buehler, Chris, Teney, Damien, Johnson, Mark, Gould, Stephen, and Zhang, Lei. Bottom-up and top-down attention for image captioning and visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 6077–6086.
- [10] Andreas, Jacob, and Klein, Dan. Reasoning About Pragmatics with Neural Listeners and Speakers. *Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2016).

- [11] Andreas, Jacob, Rohrbach, Marcus, Darrell, Trevor, and Klein, Dan. Neural module networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
- [12] Antol, Stanislaw, Agrawal, Aishwarya, Lu, Jiasen, Mitchell, Margaret, Batra, Dhruv, Zitnick, C. Lawrence, and Parikh, Devi. VQA: Visual Question Answering. In *IEEE International Conference on Computer Vision (ICCV)* (2015).
- [13] Bajaj, Mohit, Wang, Lanjun, and Sigal, Leonid. GraphGround: Graph-Based Language Grounding. In *International Conference on Computer Vision (ICCV)* (2019).
- [14] Bajaj, Mohit, Wang, Lanjun, and Sigal, Leonid. GraphGround: Graph-Based Language Grounding. In *IEEE International Conference on Computer Vision (ICCV)* (2019).
- [15] Bajcsy, Ruzena. Computer description of textured surfaces. In *Proceedings of the 3rd international joint conference on Artificial intelligence* (1973), pp. 572–579.
- [16] Banerjee, Satanjeev, and Lavie, Alon. 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, June 2005), Association for Computational Linguistics, pp. 65–72.
- [17] Bell, Sean, Upchurch, Paul, Snaveley, Noah, and Bala, Kavita. Material recognition in the wild with the materials in context database. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 3479–3487.
- [18] Berg, Tamara L, Berg, Alexander C, and Shih, Jonathan. Automatic attribute discovery and characterization from noisy web data. In *European Conference on Computer Vision (ECCV)* (2010).
- [19] Bhushan, Nalini, Rao, A Ravishankar, and Lohse, Gerald L. The texture lexicon: Understanding the categorization of visual texture terms and their relationship to texture images. *Cognitive Science* 21, 2 (1997), 219–246.
- [20] Bojanowski, Piotr, Grave, Edouard, Joulin, Armand, and Mikolov, Tomas. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.
- [21] Brendel, Wieland, and Bethge, Matthias. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *International Conference on Learning Representations (ICLR)* (2019).
- [22] Changpinyo, Soravit, Sharma, Piyush, Ding, Nan, and Soricut, Radu. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 3558–3568.

- [23] Chen, Ding-Jie, Jia, Songhao, Lo, Yi-Chen, Chen, Hwann-Tzong, and Liu, Tyng-Luh. See-through-text grouping for referring image segmentation. In *International Conference on Computer Vision (ICCV)* (2019).
- [24] Chen, H., Gallagher, A., and Girod, B. Describing clothing by semantic attributes. In *European Conference on Computer Vision (ECCV)* (2012).
- [25] Chen, Kan, Kovvuri, Rama, and Nevatia, Ram. Query-guided regression network with context policy for phrase grounding. In *IEEE International Conference on Computer Vision (ICCV)* (2017).
- [26] Chen, Mark, Radford, Alec, Child, Rewon, Wu, Jeffrey, Jun, Heewoo, Luan, David, and Sutskever, Ilya. Generative pretraining from pixels. In *International Conference on Machine Learning* (2020), PMLR, pp. 1691–1703.
- [27] Chen, Yen-Chun, Li, Linjie, Yu, Licheng, El Kholy, Ahmed, Ahmed, Faisal, Gan, Zhe, Cheng, Yu, and Liu, Jingjing. Uniter: Learning universal image-text representations.
- [28] Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , and Vedaldi, A. Describing textures in the wild, 2014.
- [29] Cimpoi, Mircea, Maji, Subhransu, and Vedaldi, Andrea. Deep filter banks for texture recognition and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).
- [30] Das, Abhishek, Kottur, Satwik, Gupta, Khushi, Singh, Avi, Yadav, Deshraj, Moura, José M. F., Parikh, Devi, and Batra, Dhruv. Visual dialog. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).
- [31] Datta, Samyak, Sikka, Karan, Roy, Anirban, Ahuja, Karuna, Parikh, Devi, and Divakaran, Ajay. Align2ground: Weakly supervised phrase grounding guided by image-caption alignment. In *IEEE International Conference on Computer Vision (ICCV)* (2019).
- [32] de Vries, Harm, Strub, Florian, Chandar, Sarath, Pietquin, Olivier, Larochelle, Hugo, and Courville, Aaron. Guesswhat?! visual object discovery through multi-modal dialogu. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).
- [33] Deng, Chaorui, Wu, Qi, Wu, Qingyao, Hu, Fuyuan, Lyu, Fan, and Tan, Mingkui. Visual grounding via accumulated attention. In *Computer Vision and Pattern Recognition (CVPR)* (2018).
- [34] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09* (2009).

- [35] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [36] Devlin, Jacob, Gupta, Saurabh, Girshick, Ross, Mitchell, Margaret, and Zitnick, C Lawrence. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467* (2015).
- [37] Dogan, Pelin, Sigal, Leonid, and Gross, Markus. Neural sequential phrase grounding (seqGROUND). In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
- [38] Donahue, Jeffrey, Anne Hendricks, Lisa, Guadarrama, Sergio, Rohrbach, Marcus, Venugopalan, Subhashini, Saenko, Kate, and Darrell, Trevor. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).
- [39] Dosovitskiy, Alexey, Beyer, Lucas, Kolesnikov, Alexander, Weissenborn, Dirk, Zhai, Xiaohua, Unterthiner, Thomas, Dehghani, Mostafa, Minderer, Matthias, Heigold, Georg, Gelly, Sylvain, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).
- [40] Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009).
- [41] Farhadi, Ali, Endres, Ian, and Hoiem, Derek. Attribute-centric recognition for cross-category generalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010).
- [42] Galatolo, Federico A, Cimino, Mario GCA, and Vaglini, Gigliola. Generating images from caption and vice versa via clip-guided generative latent space search. *arXiv preprint arXiv:2102.01645* (2021).
- [43] Gao, Peng, Jiang, Zhengkai, You, Haoxuan, Lu, Pan, Hoi, Steven CH, Wang, Xiaogang, and Li, Hongsheng. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 6639–6648.
- [44] Geirhos, Robert, Rubisch, Patricia, Michaelis, Claudio, Bethge, Matthias, Wichmann, Felix A, and Brendel, Wieland. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations (ICLR)* (2018).
- [45] Grice, H Paul. Logic and conversation.
- [46] He, Kaiming, Gkioxari, Georgia, Dollár, Piotr, and Girshick, Ross B. Mask R-CNN. In *International Conference on Computer Vision ICCV* (2017).

- [47] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
- [48] Hochreiter, Sepp, and Schmidhuber, Jürgen. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [49] Hoffer, Elad, and Ailon, Nir. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition* (2015), Springer, pp. 84–92.
- [50] Hosseini, Hossein, Xiao, Baicen, Jaiswal, Mayoore, and Poovendran, Radha. Assessing shape bias property of convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2018).
- [51] Hu, Ronghang, Rohrbach, Marcus, Andreas, Jacob, Darrell, Trevor, and Saenko, Kate. Modeling relationships in referential expressions with compositional modular networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).
- [52] Hu, Ronghang, Rohrbach, Marcus, and Darrell, Trevor. Segmentation from natural language expressions. In *European Conference on Computer Vision (ECCV)* (2016).
- [53] Hu, Ronghang, Xu, Huazhe, Rohrbach, Marcus, Feng, Jiashi, Saenko, Kate, and Darrell, Trevor. Natural language object retrieval. *Computer Vision and Pattern Recognition (CVPR)* (2016).
- [54] Huo, Yuqi, Zhang, Manli, Liu, Guangzhen, Lu, Haoyu, Gao, Yizhao, Yang, Guoxing, Wen, Jingyuan, Zhang, Heng, Xu, Baogui, Zheng, Weihao, et al. Wenlan: Bridging vision and language by large-scale multi-modal pre-training. *arXiv preprint arXiv:2103.06561* (2021).
- [55] Ioffe, Sergey, and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning* (2015).
- [56] Izadinia, Hamid, Sadeghi, Fereshteh, Divvala, Santosh K, Hajishirzi, Hannaneh, Choi, Yejin, and Farhadi, Ali. Segment-phrase table for semantic segmentation, visual entailment and paraphrasing. In *IEEE International Conference on Computer Vision (ICCV)* (2015).
- [57] Jayaraman, Dinesh, and Grauman, Kristen. Zero-shot recognition with unreliable attributes. In *Advances in Neural Information Processing Systems* (2014).
- [58] Jia, Chao, Yang, Yinfei, Xia, Ye, Chen, Yi-Ting, Parekh, Zarana, Pham, Hieu, Le, Quoc V, Sung, Yunhsuan, Li, Zhen, and Duerig, Tom. Scaling up visual and vision-language representation learning with noisy text supervision. *arXiv preprint arXiv:2102.05918* (2021).

- [59] Johnson, Justin, Hariharan, Bharath, van der Maaten, Laurens, Fei-Fei, Li, Zitnick, C. Lawrence, and Girshick, Ross. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).
- [60] Kazemzadeh, Sahar, Ordonez, Vicente, Matten, Mark, and Berg, Tamara L. Refer-ItGame: Referring to objects in photographs of natural scenes. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014).
- [61] Kim, Jin-Hwa, Jun, Jaehyun, and Zhang, Byoung-Tak. Bilinear attention networks. In *Advances in Neural Information Processing Systems* (2018), pp. 1564–1574.
- [62] Kingma, Diederik, and Ba, Jimmy. ADAM: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)* (2015).
- [63] Kiros, Ryan, Salakhutdinov, Ruslan, and Zemel, Richard S. Unifying visual-semantic embeddings with multimodal neural language models. *Transactions of the Association for Computational Linguistics (TACL)* (2015).
- [64] Kovashka, A., Parikh, D., and Grauman, K. WhittleSearch: Image search with relative attribute feedback. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012).
- [65] Krishna, Ranjay, Zhu, Yuke, Groth, Oliver, Johnson, Justin, Hata, Kenji, Kravitz, Joshua, Chen, Stephanie, Kalantidis, Yannis, Li, Li-Jia, Shamma, David A, Bernstein, Michael, and Fei-Fei, Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations.
- [66] Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (2012).
- [67] Lampert, Christoph H, Nickisch, Hannes, and Harmeling, Stefan. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 36, 3 (2014), 453–465.
- [68] Li, Ruiyu, Li, Kai-Can, Kuo, Yi-Chun, Shu, Michelle, Qi, Xiaojuan, Shen, Xiaoyong, and Jia, Jiaya. Referring image segmentation via recurrent refinement networks. In *Computer Vision and Pattern Recognition (CVPR)* (2018).
- [69] Lin, Chin-Yew. Rouge: A package for automatic evaluation of summaries. p. 10.
- [70] Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)* (2014).
- [71] Lin, Tsung-Yu, and Maji, Subhransu. Visualizing and Understanding Deep Texture Representations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 2791–2799.

- [72] Lin, Tsung-Yu, RoyChowdhury, Aruni, and Maji, Subhransu. Bilinear Convolutional Neural Networks for Fine-grained Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 40, 6 (2018), 1309–1322.
- [73] Liu, Chenxi, Lin, Zhe, Shen, Xiaohui, Yang, Jimei, Lu, Xin, and Yuille, Alan. Recurrent multimodal interaction for referring image segmentation. In *International Conference on Computer Vision (ICCV)* (2017).
- [74] Liu, Daqing, Zhang, Hanwang, Wu, Feng, and Zha, Zheng-Jun. Learning to assemble neural module tree networks for visual grounding. In *International Conference on Computer Vision (ICCV)* (2019).
- [75] Liu, Jingyu, Wang, Liang, and Yang, Ming-Hsuan. Referring expression generation and comprehension via attributes. In *IEEE International Conference on Computer Vision (ICCV)* (2017).
- [76] Liu, Xihui, Wang, Zihao, Shao, Jing, Wang, Xiaogang, and Li, Hongsheng. Improving referring expression grounding with cross-modal attention-guided erasing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
- [77] Luo, Ruotian, and Shakhnarovich, Gregory. Comprehension-guided referring expressions. *Computer Vision and Pattern Recognition (CVPR)* (2017).
- [78] Maji, Subhransu. Discovering a lexicon of parts and attributes. In *Workshop on Parts and Attributes, European Conference on Computer Vision (ECCV)* (2012).
- [79] Maji, Subhransu. A taxonomy of part and attribute discovery techniques. In *Visual Attributes*. Springer International Publishing, 2017, pp. 247–268.
- [80] Maji, Subhransu, Rahtu, Esa, Kannala, Juho, Blaschko, Matthew, and Vedaldi, Andrea. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151* (2013).
- [81] Mao, Junhua, Huang, Jonathan, Toshev, Alexander, Camburu, Oana, Yuille, Alan, and Murphy, Kevin. Generation and comprehension of unambiguous object descriptions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
- [82] Margffoy-Tuay, Edgar, Pérez, Juan C, Botero, Emilio, and Arbeláez, Pablo. Dynamic multimodal instance segmentation guided by natural language queries. In *European Conference on Computer Vision (ECCV)* (2018).
- [83] Nagaraja, Varun K., Morariu, Vlad I., and Davis, Larry S. Modeling context between objects for referring expression understanding. In *European Conference on Computer Vision (ECCV)* (2016).
- [84] Nair, Vinod, and Hinton, Geoffrey E. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning* (2010).

- [85] Nilsback, Maria-Elena, and Zisserman, Andrew. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)* (Dec 2008).
- [86] Ordonez, Vicente, Kulkarni, Girish, and Berg, Tamara L. Im2text: Describing images using 1 million captioned photographs. In *Neural Information Processing Systems (NIPS)* (2011).
- [87] Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics* (2002), Association for Computational Linguistics, pp. 311–318.
- [88] Parikh, D., and Grauman, K. Relative attributes. In *IEEE International Conference on Computer Vision (ICCV)* (2011).
- [89] Parikh, Devi, and Grauman, Kristen. Interactively building a discriminative vocabulary of nameable attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2011).
- [90] Peters, Matthew E., Neumann, Mark, Iyyer, Mohit, Gardner, Matt, Clark, Christopher, Lee, Kenton, and Zettlemoyer, Luke. Deep contextualized word representations. In *Proc. of NAACL* (2018).
- [91] Plummer, Bryan A, Kordas, Paige, Hadi Kiapour, M, Zheng, Shuai, Piramuthu, Robinson, and Lazebnik, Svetlana. Conditional image-text embedding networks. In *European Conference on Computer Vision (ECCV)* (2018).
- [92] Plummer, Bryan A, Wang, Liwei, Cervantes, Chris M, Caicedo, Juan C, Hockenmaier, Julia, and Lazebnik, Svetlana. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 2641–2649.
- [93] Plummer, Bryan A., Wang, Liwei, Cervantes, Chris M., Caicedo, Juan C., Hockenmaier, Julia, and Lazebnik, Svetlana. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *International Journal of Computer Vision* 123, 1 (2017), 74–93.
- [94] Radford, Alec, Kim, Jong Wook, Hallacy, Chris, Ramesh, Aditya, Goh, Gabriel, Agarwal, Sandhini, Sastry, Girish, Askell, Amanda, Mishkin, Pamela, Clark, Jack, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020* (2021).
- [95] Radford, Alec, Wu, Jeffrey, Child, Rewon, Luan, David, Amodei, Dario, Sutskever, Ilya, et al. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

- [96] Reed, Scott, Akata, Zeynep, Lee, Honglak, and Schiele, Bernt. Learning deep representations of fine-grained visual descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
- [97] Rohrbach, Anna, Rohrbach, Marcus, Hu, Ronghang, Darrell, Trevor, and Schiele, Bernt. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision (ECCV)* (2016).
- [98] Sadeghi, Fereshteh, Kumar Divvala, Santosh K, and Farhadi, Ali. VISKE: Visual knowledge extraction and question answering by visual verification of relation phrases. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).
- [99] Sadeghi, Mohammad Amin, and Farhadi, Ali. Recognition using visual phrases. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2011).
- [100] Sadhu, Arka, Chen, Kan, and Nevatia, Ram. Zero-shot grounding of objects from natural language queries. In *International Conference on Computer Vision (ICCV)* (2019).
- [101] Scheirer, Walter J, Kumar, Neeraj, Belhumeur, Peter N, and Boulton, Terrance E. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012).
- [102] Schuster, Mike, and Paliwal, Kuldeep K. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* 45, 11 (1997), 2673–2681.
- [103] Sharma, Piyush, Ding, Nan, Goodman, Sebastian, and Soricute, Radu. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2018), pp. 2556–2565.
- [104] Shi, Hengcan, Li, Hongliang, Meng, Fanman, and Wu, Qingbo. Key-word-aware network for referring expression image segmentation. In *European Conference on Computer Vision (ECCV)* (2018).
- [105] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *CoRR abs/1409.1556* (2014).
- [106] Su, Jong-Chyi, Wu, Chenyun, Jiang, Huaizu, and Maji, Subhransu. Reasoning about fine-grained attribute phrases using reference games. In *International Conference on Computer Vision (ICCV)* (2017).
- [107] Tamura, Hideyuki, Mori, Shunji, and Yamawaki, Takashi. Textural features corresponding to visual perception. *IEEE Transactions on Systems, man, and cybernetics* 8, 6 (1978), 460–473.
- [108] Tan, Hao, and Bansal, Mohit. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490* (2019).

- [109] Turakhia, Naman, and Parikh, Devi. Attribute dominance: What pops out? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2013).
- [110] van der Maaten, Laurens, and Hinton, Geoffrey. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)* 9, Nov (2008).
- [111] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, and Polosukhin, Illia. Attention is all you need. In *Advances in Neural Information Processing Systems* (2017), pp. 5998–6008.
- [112] Vedaldi, A., Mahendran, S., Tsogkas, S., Maji, S., Girshick, B., Kannala, J., Rahtu, E., Kokkinos, I., Blaschko, M. B., Weiss, D., Taskar, B., Simonyan, K., Saphra, N., and Mohamed, S. Understanding objects in detail with fine-grained attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014).
- [113] Vedantam, Ramakrishna, Bengio, Samy, Murphy, Kevin, Parikh, Devi, and Chechik, Gal. Context-aware captions from context-agnostic supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).
- [114] Vedantam, Ramakrishna, Lawrence Zitnick, C, and Parikh, Devi. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 4566–4575.
- [115] Vinyals, Oriol, Toshev, Alexander, Bengio, Samy, and Erhan, Dumitru. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).
- [116] Wah, Catherine, Branson, Steven, Welinder, Peter, Perona, Pietro, and Belongie, Serge. The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology, 2011.
- [117] Wah, Catherine, Van Horn, Grant, Branson, Steve, Maji, Subhransu, Perona, Pietro, and Belongie, Serge. Similarity comparisons for interactive fine-grained categorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014).
- [118] Wang, Liwei, Li, Yin, and Lazebnik, Svetlana. Learning deep structure-preserving image-text embeddings. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
- [119] Wang, Peng, Wu, Qi, Cao, Jiewei, Shen, Chunhua, Gao, Lianli, and Hengel, Anton van den. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Computer Vision and Pattern Recognition (CVPR)* (2019).
- [120] Wu, Chenyun, Lin, Zhe, Cohen, Scott, Bui, Trung, and Maji, Subhransu. Phrase-cut: Language-based image segmentation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 10216–10225.

- [121] Wu, Chenyun, Timm, Mikayla, and Maji, Subhransu. Describing textures using natural language. In *Proceedings of the European Conference on Computer Vision (ECCV)* (August 2020).
- [122] Xiao, Fanyi, Sigal, Leonid, and Jae Lee, Yong. Weakly-supervised visual grounding of phrases with linguistic structures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).
- [123] Xu, Kelvin, Ba, Jimmy, Kiros, Ryan, Cho, Kyunghyun, Courville, Aaron, Salakhudinov, Ruslan, Zemel, Rich, and Bengio, Yoshua. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning* (2015), pp. 2048–2057.
- [124] Yang, Sibe, Li, Guanbin, and Yu, Yizhou. Dynamic graph attention for referring expression comprehension. In *International Conference on Computer Vision (ICCV)* (2019).
- [125] Yang, Zhengyuan, Gong, Boqing, Wang, Liwei, Huang, Wenbing, Yu, Dong, and Luo, Jiebo. A fast and accurate one-stage approach to visual grounding. In *International Conference on Computer Vision (ICCV)* (2019).
- [126] Ye, Linwei, Rochan, Mrigank, Liu, Zhi, and Wang, Yang. Cross-modal self-attention network for referring image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
- [127] Young, Peter, Lai, Alice, Hodosh, Micah, and Hockenmaier, Julia. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics 2* (2014), 67–78.
- [128] Yu, Licheng, Lin, Zhe, Shen, Xiaohui, Yang, Jimei, Lu, Xin, Bansal, Mohit, and Berg, Tamara L. MattNet: Modular attention network for referring expression comprehension. In *Computer Vision and Pattern Recognition (CVPR)* (2018).
- [129] Yu, Licheng, Poirson, Patrick, Yang, Shan, Berg, Alexander C, and Berg, Tamara L. Modeling context in referring expressions. In *European Conference on Computer Vision (ECCV)* (2016).
- [130] Yu, Zhou, Yu, Jun, Cui, Yuhao, Tao, Dacheng, and Tian, Qi. Deep modular co-attention networks for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 6281–6290.
- [131] Yu, Zhou, Yu, Jun, Xiang, Chenchao, Zhao, Zhou, Tian, Qi, and Tao, Dacheng. Rethinking diversified and discriminative proposal generation for visual grounding. *CoRR abs/1805.03508* (2018).
- [132] Zhang, Hanwang, Niu, Yulei, and Chang, Shih-Fu. Grounding referring expressions in images by variational context. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

- [133] Zhang, Peng, Goyal, Yash, Summers-Stay, Douglas, Batra, Dhruv, and Parikh, Devi. Yin and Yang: Balancing and answering binary visual questions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
- [134] Zhu, Yukun, Kiros, Ryan, Zemel, Rich, Salakhutdinov, Ruslan, Urtasun, Raquel, Torralba, Antonio, and Fidler, Sanja. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 19–27.