ABSTRACT

Title of dissertation: TOWARDS ROBUST, INTERPRETABLE AND SCALABLE VISUAL REPRESENTATIONS

Ang Li, Doctor of Philosophy, 2017

Dissertation directed by: Professor Larry S. Davis
Department of Computer Science

Visual representation is one of the central problems in computer vision. The essential problem is to develop a unified representation that effectively encodes both visual appearance and spatial information so that it can be easily applied to various vision applications such as face recognition, image matching, and multimodal image retrieval. Along with the history of computer vision research, there are four major levels of visual representations, i.e., geometric, low-level, mid-level and high-level. The dissertation comprises four works studying effective visual representations in the four different levels. Multiple approaches are proposed with the aim of improving the robustness, interpretability, and scalability of visual representations.

Geometric features are effective in matching images under spatial transformations however their performance is sensitive to the noises. In the first part, we propose to model the uncertainty of geometric representation based on line segments

and propose to equip these features with uncertainty modeling so that they could be robustly applied in the image-based geolocation application.

We study in the second part the robustness of feature encoding to noisy keypoints. We show that traditional feature encoding is sensitive to background or noisy features. We propose the Selective Encoding framework which learns the relevance distribution of each codeword and incorporate such information with the original codebook model. Our approach is more robust to the localization errors or uncertainty in the active face authentication application.

The mission of visual understanding is to express and describe the image content which is essentially relating images to human language. That typically involves finding a common representation inferable from both domains of data. In the third part, we propose a framework to extract a mid-level spatial representation directly from language descriptions and match such spatial layouts to the detected object bounding boxes for retrieving indoor scene images from user text queries.

Modern high-level visual features are typically learned from supervised datasets, whose scalability is largely limited by the requirement of dedicated human annotation. In the last part, we propose to learn visual representations from large-scale weakly supervised data for a large number of natural language-based concepts, i.e., n-gram phrases. We propose the differentiable Jelinek-Mercer smoothing loss and train a deep convolutional neural network from images with associated user comments. We show that the learned model can predict a large number of phrase-based concepts from images, can be effectively applied to image-caption applications and transfers well to other visual recognition datasets.

TOWARDS ROBUST, INTERPRETABLE AND SCALABLE VISUAL REPRESENTATIONS

by

Ang Li

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2017

Advisory Committee:
Professor Larry S. Davis, Chair/Advisor
Professor Rama Chellappa
Professor Hal Daumé III
Professor Ramani Duraiswami
Professor Thomas Goldstein

Dedication

In memory of Youde Wu. He was my English teacher in high school and he was probably the first who encouraged me to go to a graduate school.

Acknowledgments

I would like to express my gratitude to my advisor, Professor Larry S. Davis, an excellent research mentor and a remarkable person who has made the five years of my graduate study a precious and enjoyable experience. Larry provides me the opportunity to work on a very diverse set of research projects which broaden my horizon and improves my understanding of the whole area. He is patient in listening to any of my immature ideas and always came up with related literature for me to read. Whenever he finds an idea interesting, he always shows his greatest support without hesitation. I appreciate his flexibility and encouragement for me to pursue interesting directions beyond the scope of projects. Apart from research, Larry also influenced me greatly in many other different ways. He has a great personality that I eager to learn from. He has a great sense of humor and he is always trying to make our meetings relaxed and enjoyable. I feel very lucky that I have had Larry as my doctoral advisor.

I would like to thank Vlad Morariu, who has been a great research mentor during my graduate study. Without his consistent help in both formulating ideas and writing papers, the thesis would be far from being completed. Thanks are due to Professor Rama Chellappa, Professor Hal Daumé III, Professor Ramani Duraiswami and Professor Thomas Goldstein for agreeing to serve on my thesis committee and for sparing their invaluable time reviewing the manuscript. I would also like to thank Laurens van der Maaten, Armand Joulin and Allan Jabri from Facebook AI Research. A part of this thesis was conducted under the collaboration with

them during my internship. I would like to express my thankfulness to Professor Amol Deshpande and my colleagues Jin Sun, Joe Ng, Ruichi Yu and Hui Miao at the University of Maryland. All of them have provided me the greatest research collaboration and support.

During my graduate study, I have had the opportunity to study in many great graduate classes and I would like to thank all the teachers David Mount, Radu Balan, Mohammad Hajiaghayi, David Jacobs, Jimmy Lin, Rama Chellappa and Howard Elman. I was involved in several research projects and I would like to thank Yaser Yacoob and Vishal Patel for their supports and discussions in some of these projects. I am also fortunate to have had multiple brilliant research mentors from internships. I would like to thank all of them for their invaluable comments and discussions which help to frame the way I conduct and present research, including Feng Tang and Jennifer Shi from Apple, Chunhui Gu and Dar-Shyang Lee from Google, Jonghyun Choi and Hongchen Wang from Comcast Labs, Laurens van der Maaten and Armand Joulin from Facebook AI Research.

I would like to thank all of my other colleagues in the Computer Vision Lab and the CS department who has enriched my graduate life in many ways. I would also like to acknowledge help and support from the staff members of UMD CS, UMIACS, Bluecrab cluster and Deepthought cluster.

I would like to acknowledge financial support from the United States Air Force, Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory, Defense Advanced Research Projects Agency (DARPA), Office of Naval Research (ONR) and Facebook AI Research, Dean's Fellowship from University of

Maryland CS department, Graduate Research Appreciation Travel Award, ICDE NSF travel grant and CVPR Doctoral Consortium travel grant. I also acknowledge the University of Maryland supercomputing resources for computation support.

Finally, I owe my deepest thanks to my mother, father and grandparents who have always stood by me and guided me through my career, and have pulled me through against impossible odds at times, my heartfelt thanks to my wife, Zhujun, who gives me her unlimited support and encouragement, and makes my everyday life much more colorful and joyful, and also to my eight-month-old son, Sean, who brings me tons of memorable moments during the preparation of this dissertation.

It is impossible to remember all, and I apologize to those I've inadvertently left out.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

erf	Gauss error function
e	Euler's number
GPS	Global Positioning System
DEM	Digital Elevation Model
LIDAR	Light Detection and Ranging
FOV	Field of View
OCM	Oriented Chamfer Matching
DCM	Directional Chamfer Matching
DT	Distance Transform
AUC	Area Under the Curve
SVM	Support Vector Machine
MIL	Multiple Instance Learning
FV	Fisher Vector
GMM	Gaussian Mixture Model
LFW	Labeled Faces in the Wild
TPR	True Positive Rate
EER	Equal Error Rate
PCA	Principal Component Analysis
SIFT	Scale Invariant Feature Transform
YTF	Youtube Faces
ROC	Receiver Operating Characteristic
DBM	Deep Boltzman Machine
CRF	Conditional Random Field
NLP	Natural Language Processing
IOU	Intersection Over Union
NN	Nearest Neighbor
CNN	Convolutional Neural Network
GT	Ground Truth
RCNN	Region-based Convolution Neural Network
DPM	Deformable Part-based Model
GAN	Generative Adversarial Network
JM	Jenlinek-Mercer
KN	Kneser-Ney
NIC	Neural Image Captioning
CCA	Canonical Correlation Analysis
RNN	Recurrent Neural Network
NLL	Negative Log Likelihood

Chapter 1: Introduction

1.1 Motivation

Visual representation is one of the central problems in computer vision which studies how the visual information can be represented so that the representation can be easily applied to various vision applications such as object tracking [1–5], image segmentation [6–8] and video classification [9–11]. Since the early days of computer vision, probably in 1960s [12], much research effort has been focused on developing effective and efficient visual representations, most of which are driven by tasks.

From 1960s to around 1990s, a majority of work is focused on extracting geometric representations from image data such as boundary detection [13], line detection [14], early vision processing [15], chamfer matching [16], edge detection [17,18]. Geometry is an essential representation in various vision applications which inherently occurs in our visual world. The research on geometry is still continuing in the recent years, such as line segment detector [19].

Since the 1990s, research attentions were shifted a little to the low-level visual representations based on key-point feature descriptions. Renowned works include Harris corner detection [20], Scale Invariant Feature Transform (SIFT) [21], Histogram of Oriented Gradients (HOG) [22]. The central idea of these representations

is to represent an image using a set of feature key-points and represent each feature key-point using local appearance descriptors.

Following the invention of local features, many approaches are developed aiming at matching two images represented by key-point descriptors. Bag-of-visual-words [23] models first quantize the visual features into multiple clusters and represent the set of features as the histogram of quantized features. Many other works include Pyramid Match Kernels [24], Spatial Pyramid Matching [25], Deformable Part-based Models [26] and mining mid-level discriminative patches [27]. Among these works, many methods are categorized into mid-level representations which utilize both low-level features and global geometric structure of the whole image.

Since around 2010s when the first large scale image classification dataset, ImageNet [28], was created, research was speeded up on understanding the high-level semantics of the image data. While most of the existing approaches had been trained and evaluated in small datasets, the creation of large dataset brought new promising research in creating visual representations, capable of recognizing a large set of visual semantics. Driven by large scale training data, the deep learning technique has been one of the most successful methods in computer vision, the emerging of which probably starts in 2012 when AlexNet [29] won the ImageNet challenge. Lots of new neural network architectures are invented afterward such as GoogLeNet [30], VGG network [31], ResNet [32], and DenseNet [33].

To summarize, along with the history of visual representation research, there are four major categories of visual representations, i.e., geometric, low-level, mid-level and high-level representations. While there are several objectives in designing

visual representations, the thesis focuses on the following three major ones:

- robustness: the representation should be robust to the input data noise;
- interpretability: the representation should be interpretable by human;
- scalability: the representation should cover a large amount of visual semantics.

The dissertation comprises four works, motivated by different applications and aiming at improving the visual representation in terms of robustness, interpretability, and scalability. The next section gives an overview introduction to each of the four approaches separately.

1.2 Approaches

Probabilistic geometric representation for cross-view matching. Cross-view or cross-pose image matching is seen as one of the most challenging problems in computer vision. An interesting application of such is the image-based geolocation task, which aims at matching a ground-based photo to a large set of satellite images in order to estimate the geolocation of the photo. It is essentially a cross-view image matching but what makes it much more difficult is the large projective transformation and the color discrepancy between the two domains of image data. Considering that, we choose to use the geometric structure as the feature representation which can be easily transformed with projection. Geometry is probably the lowest level representation that one can use, which is sufficiently simplified to reduce the effect of color discrepancy issues. However, their performance is sensitive to the noises, especially

when the noise can be amplified under the non-linear projective transformation. So we propose to model the uncertainty of such linear geometric structure and propose a probabilistic geometric representation that is incorporated with the uncertainty modeling. We show such probabilistic representation is more robust to line segment annotation or detection noises.

Robust encoding of low-level features for mobile face authentication. Keypoint based visual features have been studied for decades and shown their strength in many low-level vision problems especially image matching and retrieval. Bag-of-words or codebook models are invented as a way to encode a set of extracted low-level features into a compact feature representation, based on feature quantization (clustering) and feature assignment. Improvements are made in terms of how the internal statistics of each feature cluster is represented. The most successful ones are vector of locally aggregated descriptors (VLAD) [34] and Fisher vector encoding [35]. We look at the application of mobile-phone based active face authentication where the real-time recorded face images are matched to a constrained set of pre-recorded user faces in order to authenticate the current users. However, these face images are often under very different poses due to the limited viewpoint of the front cameras of smartphones. That leads to a high failure rate of pre-trained face detectors. Considering also the computational power of smartphones, we follow a popular pipeline that uses densely sampled SIFT features and Fisher vector encoding to represent every incoming image. However, such approach suffers from severe sensitivity to background noises. To address this issue, we propose to learn a probability distri-

bution of each cluster in the codebook with respect to the relevance of features to the facial area. Based on such relevance estimation, we inject a new selector component into the codebook pipeline so that the final feature encoding becomes more robust to the distractions from the background area.

Mid-level spatial layout representation for multimodal matching. The mission of visual understanding is to express and describe the image content which is essentially relating images to human language. That typically involves finding a common representation inferable from both domains of data. We propose a framework to extract a mid-level spatial representation directly from language descriptions. Such spatial representation is based on solving a mathematical programming to sample the object bounding boxes in 3D world space and further projected into the 2D image space. We match such spatial layouts to the detected object bounding boxes for retrieving indoor scene images from user text queries. Our approach is inspired by human's capability of generating vague scenes in their own mind without any reference to visual inputs. While such generative modeling process is still challenging, we provide an alternative way to address the multimodal matching problem by showing that our solver is able to generate feasible abstract scenes from language only and that the image retrieval success rate can be significantly improved based on matching these sampled abstractions.

Learning large scale high-level visual representations of phrases. Modern high-level visual features are typically learned from supervised datasets, for example, the Im-

ageNet challenge dataset. However, the scalability of these approaches is largely limited by the requirement of dedicated human annotation. That is, the supervised model becomes difficult to generalize to many other real world concepts which are not necessarily included in the training categories. In order to increase the model's capability, more human annotations are needed which could be expensive and sometimes even infeasible to collect considering the scale of real world concepts (an estimation on the number of phrases in English would be over several million). Motivated by that, we propose to explore weakly supervised approaches. We propose to learn visual representations from large-scale social media data for a large number of natural language-based concepts, i.e., n-gram phrases. We use a feedforward Convolutional Neural Network to learn from images with associated user comments. By noticing that the naive softmax loss function does not necessarily make use of the relationships between different n-grams and does not handle the out-of-vocabulary phrases, we propose the differentiable Jelinek-Mercer smoothing loss and train a deep convolutional neural network using such loss. The Jelinek-Mercer loss function is inspired by the Jelinek-Mercer smoothing technique in the conventional n-gram language modeling. We show the proposed Jelinek-Mercer loss significantly outperforms traditional softmax loss function and we show that the proposed feedforward network can be efficiently trained and effectively applied to several high-level multimodal applications such as phrase prediction, phrase-based image retrieval, caption prediction, sentence-based image retrieval and transfer learning. The resulted model captures a much wider range of visual concepts compared to traditional supervised models learned from constrained datasets.

Previous publication. The material in this thesis has been published at top-tier venues on computer vision. The probabilistic geometric representation for cross-view matching was published in *European Conference on Computer Vision* in 2014 [36]. The robust encoding of low-level features for mobile-phone active face authentication was published in *International Conference on Computer Vision* in 2015 [37]. The mid-level spatial representation extracted from the language for multimodal matching appeared in *IEEE Conference on Computer Vision and Pattern Recognition* in 2017 [38]. The work for learning large scale high-level visual representations of phrases is accepted in *International Conference on Computer Vision*, 2017 [39,40].

1.3 Organization

The thesis is organized as follows. Chapter 2 introduces the probabilistic geometric representation for cross-view matching. Chapter 3 introduces the robust encoding of low-level features for mobile-phone active face authentication. Chapter 4 introduces the mid-level spatial layout representation extracted from the language for multimodal matching. Chapter 5 introduces learning large scale high-level visual representations of phrases. The thesis is then concluded in Chapter 6. Additional theoretical proofs in the image geolocation work is shown in Appendix A. Additional results for text-based image retrieval using the abstract spatial layout representation are shown in Appendix B. Additional results for the visual n-gram models are shown in Appendix C. License information for images from YFCC100M dataset used in this thesis is detailed in Appendix D.

Chapter 2: Representing linear geometry under projective uncertainty

2.1 Motivation

Figure 2.1: Geolocation involves finding the corresponding location of the ground image (on the left) in ortho images (an example on the right).

Given a ground-level photograph, the image geolocation task is to estimate the geographic location and orientation of the camera (Figure 2.1). Such systems provide an alternative way to localize an image or a scene when and where GPS is unavailable. Visual based geolocation has wide applications in areas such as robotics, autonomous driving, news image organization and geographic information systems. We focus on a single image geolocation task which compares a single ground-based

query image against a database of ortho images over the candidate geolocations. Each of the candidate ortho images is evaluated and ranked according to the query. This task is difficult because (1) significant color discrepancy exists between cameras used for ground and ortho images; (2) the images taken at different times result in appearance difference even for the same locations (e.g. a community before and after being developed); (3) the ortho image databases usually have a very large scale, which requires efficient algorithms.

Due to the difficulty of the geolocation problem, many recent works include extra data such as georeferenced image databases [41, 42], digital elevation models (DEM) [43], light detection and ranging (LIDAR) data [44], etc. Whenever photographs need to be geolocated in a new geographic area, this side data has to be acquired first. This limits the expandability of these geolocation approaches. One natural question to ask is whether we can localize a ground photograph using only widely accessible satellite images.

We address this geolocation task with no side data by casting it as an image matching problem. This is challenging because the camera orientation of a ground image is approximately orthogonal to that of its corresponding ortho image. Commonly used image features are not invariant to such wide camera rotation. In addition, considering the presence of color and lighting difference between ground and ortho images, color-based and intensity-based image features become unreliable for establishing image correspondence. Therefore, structural information becomes the most feasible feature for this application. We utilize linear structures – line segments – as the features to be matched between ground and ortho images.

Both ground and ortho images are projections of the 3D world. The information loss between these two images becomes an obstacle even for matching binary line segments. Instead of inferring 3D structure, we extract and match the linear structures that lie on the ground a large subset of which is visible in both ground and ortho images. The ortho images can be regarded as approximately 2D planes and we use classic line extraction algorithms to locate the extended linear structures in them. The ground images are more challenging so we ask humans annotate the ground lines for these images. This is not a burdensome task. Additionally, the horizon line is annotated by the human so we can construct its corresponding aerial view with the camera parameters known.

Based on chamfer matching [45], we derive a criterion function for matching each ortho image with the ortho-rectified view of the ground image. However, the projection matrix for transforming the ground image to its ortho view is usually numerically ill-conditioned. Even a small perturbation to the annotated end points of a line segment may result in significant uncertainty in location and orientation of the projected line segments, especially those near the horizon line. Therefore, we propose a probablistic representation of line segments by modeling their uncertainty and introduce a model of geometric uncertainty into our matching criterion. Within each ortho image, the matching scores for possible pairs of camera locations and orientations are exhaustively evaluated. This sliding window search is speeded up by means of distance transforms [46] and convolution operations.

Contribution. The main contributions of this work include

- An uncertainty model for line segments under projective transformations;
- A novel distance transform based matching criterion under uncertainty;
- The application of geometric matching to single image geolocation with no side data.

2.2 Related work

2.2.1 Image geolocation

Previous work on image geolocation can be classified into two main streams: geotagged image retrieval and model based matching. Hays et al. [41] were among the first to treat the image geolocation as a data driven image retrieval problem. Their approach is based on a large scale geotagged image database. Those images with similar visual appearance to the query image are extracted and their GPS tags are collected to generate a confidence map for possible geolocations. Li et al. [47] devised an algorithm to match low level features from large scale database to ground image features in a prioritized order specified by likelihood. Similar approaches improve the image retrieval algorithms applied to ground level image databases [48–51]. Generally, data driven approaches assume all possible views of the ground images are covered in the database. Otherwise, the system will not return a reasonable geolocation.

Apart from the retrieval-style geolocation, the other track is to match the image geometry with 3D models to estimate the camera pose. Battz et al. [43]

proposed a solution to address the geolocation in mountainous terrain area by extracting skyline contours from ground images and matching them to the digital elevation models. From the 3D reconstruction viewpoint, some other approaches estimate the camera pose by matching images with 3D point cloud [52–54].

Few works make use of the satellite images in the geolocation task. Bansal et al. [55] match the satellite images and aerial images by finding the facade of the building and rectifying the facade for matching with the query ground images. Lin et al. [42] address the out-of-sample generalization problem suffered by data-driven methods. The core of their method is learning a cross-view feature correspondence between ground and ortho images. However, their approach still requires a considerable amount of geo-tagged image data for learning.

Our work differs from all of the above work in that our approach casts the geolocation task as a linear geometric matching problem instead of reconstructing the 3D world, and it is relatively "low-cost" using only the satellite images without the need for large labeled training sets or machine learning.

2.2.2 Geometric matching

In the geometric matching domain, our approach is related to line matching and shape matching. Matching line segments has been an important problem in geometric modeling. Schmid et al. [56] proposed a line matching approach based on cross correlation of neighborhood intensity. This approach is limited by its requirement on prior knowledge of the epipolar geometry. Bay et al. [57] match line

segments using color histograms and remove false correspondences by topological filtering. In recent years, line segments have been shown to be robust to matching images in poorly textured scenes [58, 59]. Most of the existing works rely on local appearance-based features while our approach is completely based on matching the binary linear structures.

Our approach is motivated by chamfer matching [60], which has been widely applied in shape matching. Chamfer matching involves finding for each feature in an image its nearest feature in the other image. The computation can be efficiently achieved via distance transforms. A natural extension of chamfer matching is to incorporate the point orientation as an additional feature. Shotton et al. [61] proposed oriented chamfer matching by adding an angle difference term into their formulation and applied this technique in matching contour fragments for general object recognition. Another method for encoding the orientation is the fast directional chamfer matching proposed by Liu et al. [45]. They generalize the original chamfer matching approach by seeing each point as a 3D feature which is composed of both location and orientation. Efficient algorithms are employed for computing the 3D distance transform based on [46]. However, for geolocation, our problem is to match a small linear structures to fairly large structures that contain much noise, especially in ortho images. Our approach is carefully designed specifically for the needs of geolocation: it takes into account the projective transformations and line segments with uncertain end points as part of the matching criterion function.

2.2.3 Uncertainty modeling

Uncertainty is often involved in various computer vision problems. Olson [62] proposed a probabilistic formulation for Hausdorff matching. Similar to Olsons work, Elgammal et al. [63] extended Chamfer matching to a probabilistic formulation. Both approaches consider only the problem of matching an exact model to uncertain image features, while our work handles the situation when the model is uncertain. An uncertainty model is proposed in [64] for projective transformations in multi-camera object tracking. They considered the case where the imaged point is sufficiently far from the line at infinity and provided an approximation method to compute the uncertainty under projective transformation. Our work differs in that (1) we provide an exact solution for projective uncertainty of line segments, and (2) we do not assume that line segments are far from the horizon line. To our knowledge, none of the previous work in geolocation were incorporated with uncertainty models.

2.3 Assumptions

A query consists of a single ground image with unknown location and orientation is provided. This ground image is then matched exhaustively to each candidate ortho images, and ortho images are ranked according to their matching scores. The ortho images are densely sampled by overlapped sliding windows over the candidate geographic areas. The scale of each ortho image can be around 10 centimeters per pixel. The ground images could be taken at any location within ortho images. Even

in a 640×640 ortho image, there are over millions of possible discretized camera poses. The geolocation task is to localize the ground image into the ortho images, not necessarily the camera pose.

We have two assumptions here to simplify this problem. First, the camera parameter (focal length) for ground images is known, a reasonable assumption, since modern cameras store this information as part of the image metadata. Second, we assume the photographer holds the camera horizontally, i.e. the camera optical axis is approximately parallel to the ground. Camera rotation around the optical axis may happen and is handled by our solution. No restrictions assumed for the satellite cameras as long as satellite imagery is rectified to ensure linear structures remain linear, which is generally true.

2.4 Preprocessing

We reconstruct the aerial view of the ground image by estimating the perspective camera model from the manually annotated horizon line. In our matching approach, line segments are matched between ground and ortho images. Lines on the ground are most likely to be viewed in both ground and ortho images – most other lines are on the vertical surfaces that are not visible in satellite imagery – so we ask users to annotate only line segments on the ground plane in query images. Once the projection matrix is known, the problem becomes one of geometric matching between two planes.

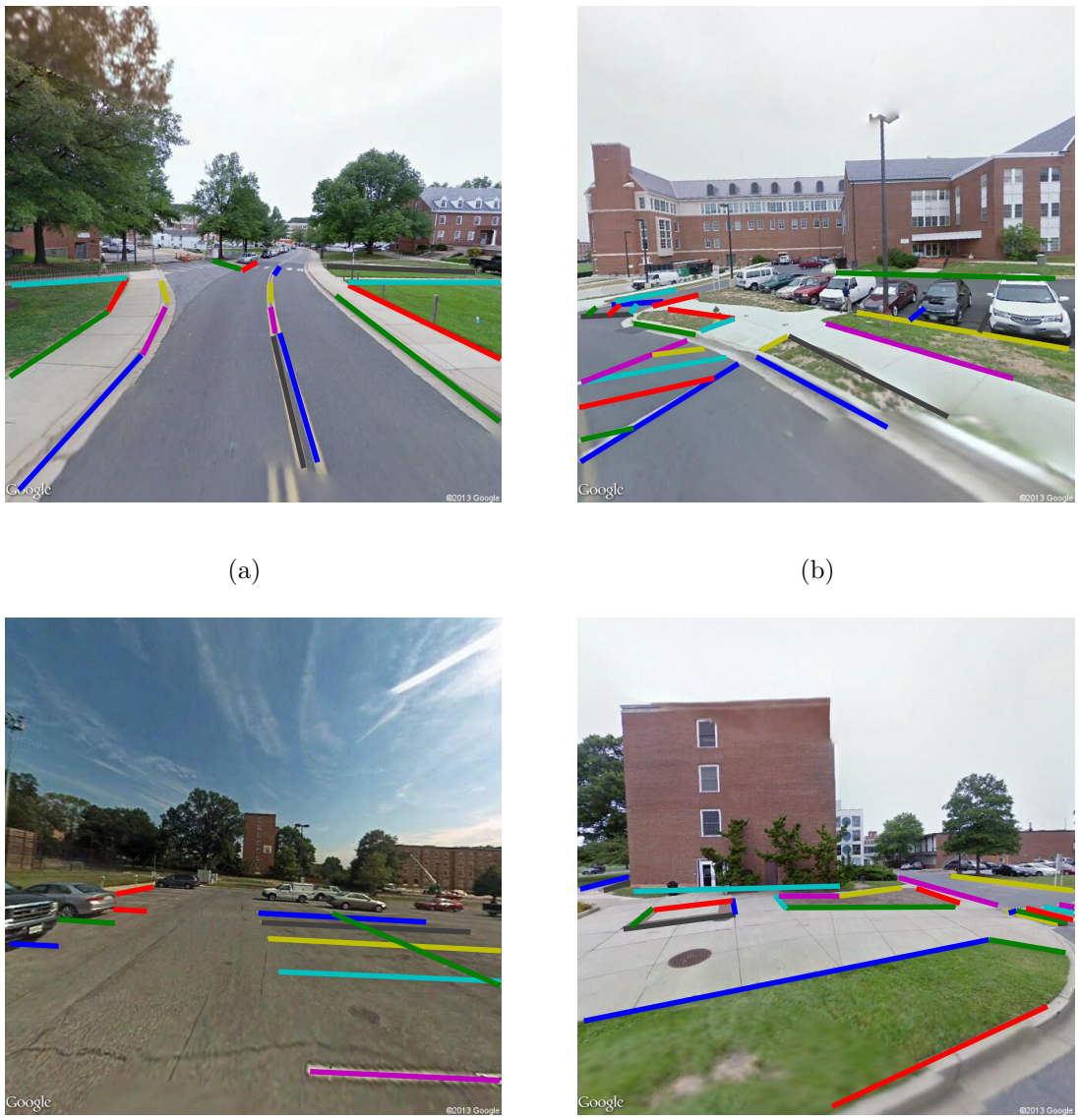
2.4.1 Line segment labeling

Line segments in ground images are annotated by human users clicking pairs of ending points. It is affordable to incorporate such human labeling process into our geolocation solution since the annotation is inexpensive and each query image needs to be labeled only once. A person can typically annotate a query image in at most two minutes. Figure 2.2 shows four ground image samples with superimposed annotated line segments.

Line segments in the ortho images are automatically detected using the approach of [65] (Figure 2.3). The detected line segments lie mostly on either the ground plane or some plane parallel to the ground, such as the roof of a building. We do not attempt to remove these non-ground lines. In fact, some of the non-ground plane lines prove useful for matching. For example, the rooflines of many buildings have the same geometry as their ground footprints. Human annotators label linear features around the bottoms of these buildings. Thus, the line segments lying on the edges of a building roof still contribute to the structure matching. Our geometric matching algorithm assumes a high level of outliers, so even if the rooflines and footprints are different the matching can still be successful.

2.4.2 Aerial view recovery

Using the computed perspective camera model, we transform the delineated ground photo line segments to an overhead view (Figure 2.4). Two assumptions are made for recovering the aerial view from ground images: (1) the camera focal length



(a) (b)

(c) (d)

Figure 2.2: Examples of line segments annotated in ground images.

f is known, and (2) the optical axis of camera is parallel to the ground plane, i.e. the camera is held horizontally. These assumptions are not sufficient for reconstructing a complete 3D model but is sufficient for recovering the ground plane given the human annotated horizon line. The horizon line is located by finding two vanishing

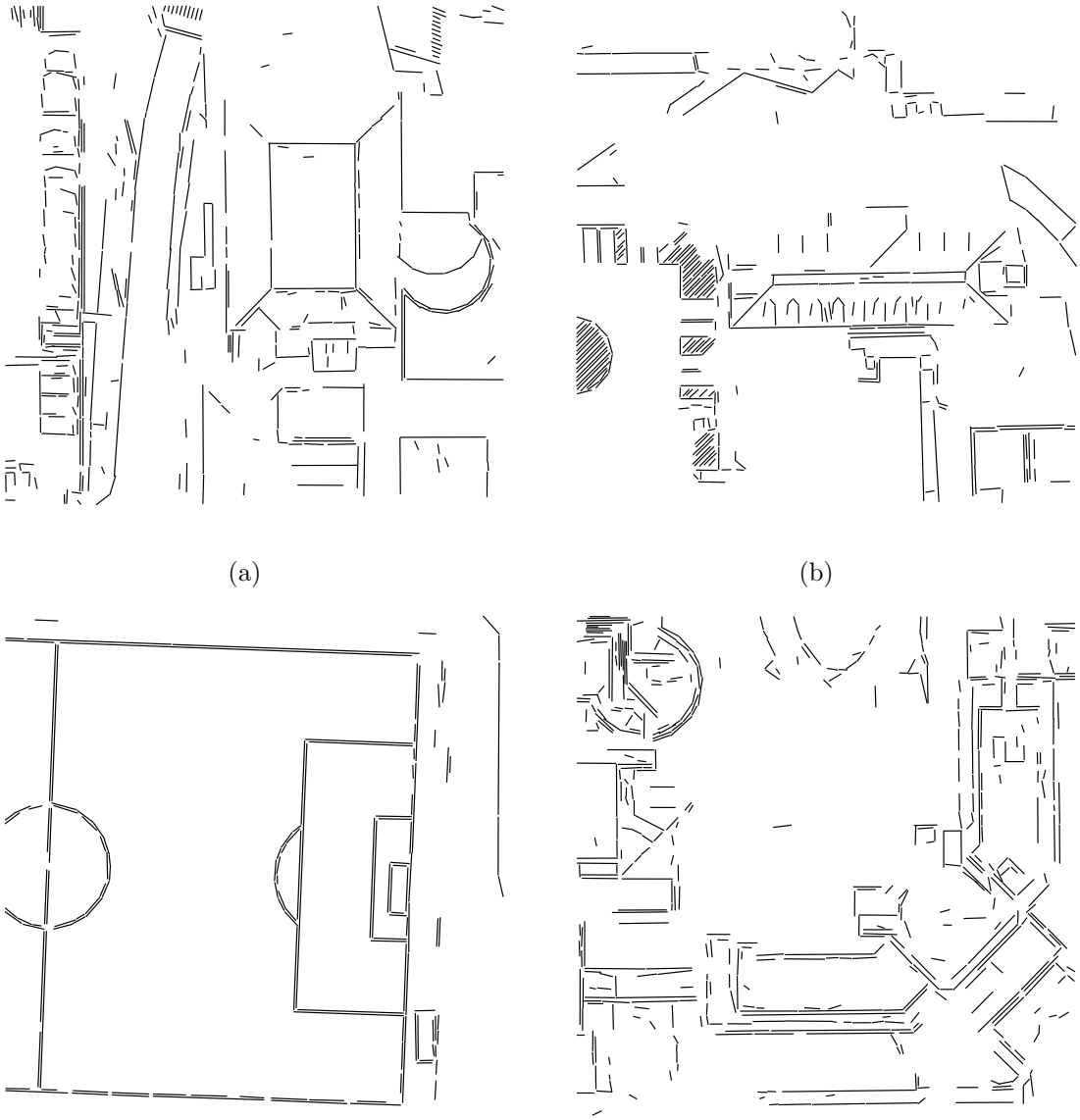


Figure 2.3: Examples of line segments detected in ortho images.

points, i.e. intersections of lines parallel in the real world.

Assuming the horizon line has slope angle θ, the ground image can be rotated clockwise by θ so that the horizon line becomes horizontal (the y-coordinate of rotated horizon line y'_0). The rotated coordinates are $(x', y')^\top = \mathbf{R}_\theta(x_g, y_g)^\top$ for

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

Figure 2.4: Ortho view recovery: (a) the original ground image where the red line is the horizon line and the blue line is shifted 50 pixels below the red line so that the ortho-rectified view will not be too large. The blue line corresponds to the top line in the converted view (c); (b) is the same image with superimposed ground line segments; (c) is the ortho-rectified view; (d) is the corresponding linear features transformed to aerial view with field of view shown by dashed lines. The field of view (FOV) is 100 degrees which can be computed according to the focal length. ©Google

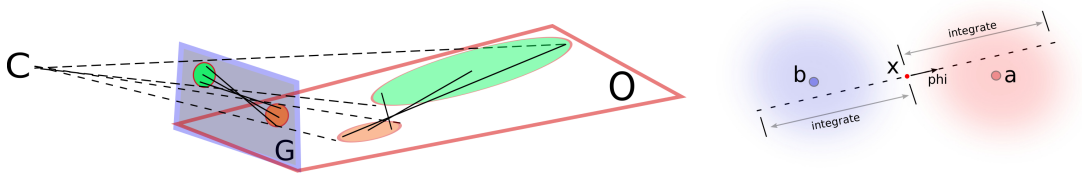
every pixel (x_g, y_g) in the original ground image. In the world coordinate system (X, Y, Z), the camera is at the origin, facing the positive direction of the Y-axis, and the ground plane is $Z = -Z_0$. If we know pixel (x', y') is on the ground, then its corresponding world location can be computed by

$$x' = \frac{fX}{Y}, y' - y'_0 = \frac{fZ_0}{Y} \Rightarrow X = \frac{x'Z_0}{y' - y'_0}, Y = \frac{fZ_0}{y' - y'_0} \ . \tag{2.1}$$

For the ortho image, a pixel location (x_o, y_o) can be converted to world coordinates by $(X, Y) = (x_o/s, y_o/s)$ where s is a scale factor with unit 1/meter relating the pixel distance to real world distance.

2.5 Uncertainty modeling for line segments

User annotations on ground images are often noisy. The two hand-selected end points could easily be misplaced by a few pixels. However, after projective transfor-



(a) (b)

Figure 2.5: (a) G is the ground image, O is the ortho-view and C is the camera. The projection from G to O results in dramatic uncertainty (b) Let a and b are centers of normal distributions. If pixel location x and the slope angle φ of the line it lies on are known, then the two end points must be on the alternative directions starting from x.

mation, even a small pertubation of one pixel can result in significant uncertainty in the location and orientation of the line segment, especially if that pixel is close to the horizon (see Figure 2.5(a)). Therefore, before discussing the matching algorithm, we first study the problem of modeling the uncertainty of line segments under projective transformation to obtain a principled probabilistic description for ground based line segments. We obtain a closed form solution by assuming that the error of labeling an end point on ground images be described by a normal distribution in the original image.

We first introduce a lemma which is essentially the integration of Gaussian density functions over a line segment.

Lemma 2.1. *Let* $\mathbf{a}, \mathbf{b}$ *be column vectors in* $\mathbb{R}^n$ *and* $\|\mathbf{a}\| = 1$*, then*

$$\int_{t_1}^{t_2} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\|\mathbf{a}t+\mathbf{b}\|^2}{2\sigma^2}} dt = e^{-\frac{\|\mathbf{b}\|^2-(\mathbf{a}^\top\mathbf{b})^2}{2\sigma^2}} \cdot \frac{1}{2}\left(erf\left(\frac{t_2+\mathbf{a}^\top\mathbf{b}}{\sqrt{2}\sigma}\right) - erf\left(\frac{t_1+\mathbf{a}^\top\mathbf{b}}{\sqrt{2}\sigma}\right)\right) \tag{2.2}$$

The proof of this lemma can be found in Appendix. Using this lemma, we derive our main theorem about uncertainty modeling. A visualization of the high level idea is shown in Figure 2.5(b).

Theorem 2.1. *Let ℓ be a 2D line segment whose end points are random variables drawn from normal distributions $N(\mathbf{a}, \sigma^2)$ and $N(\mathbf{b}, \sigma^2)$ respectively. Then for any point $\mathbf{x}$, the probability that $\mathbf{x}$ lies on ℓ and ℓ has slope angle φ is*

$$\begin{aligned} p(\mathbf{x}, \varphi|\mathbf{a}, \mathbf{b}) =& e^{-\frac{\|\mathbf{x}-\mathbf{a}\|^2 - |\langle \mathbf{x}-\mathbf{a}, \Delta_\varphi \rangle|^2 + \|\mathbf{x}-\mathbf{b}\|^2 - |\langle \mathbf{x}-\mathbf{b}, \Delta_\varphi \rangle|^2}{2\sigma^2}} \\ & \cdot \frac{1}{2}\left(1 - erf\left(\frac{\langle \mathbf{x}-\mathbf{a}, \Delta_\varphi \rangle}{\sqrt{2}\sigma}\right) erf\left(\frac{\langle \mathbf{x}-\mathbf{b}, \Delta_\varphi \rangle}{\sqrt{2}\sigma}\right)\right) \end{aligned} \tag{2.3}$$

where $\Delta_\varphi = (\cos\varphi, \sin\varphi)^\top$ is the unit vector with respect to the slope angle φ.

Proof. Let $p_n(\mathbf{x}; \boldsymbol{\mu}, \sigma^2)$ be the probability density function for normal distribution $N(\boldsymbol{\mu}, \sigma^2)$. The probability that $\mathbf{x}$ lies on the line segment equals the probability that random variables of the two ending points are $\mathbf{x} + t_a\Delta_\varphi$ and $\mathbf{x} + t_b\Delta_\varphi$ for some $t_a, t_b \in \mathbb{R}$ and $t_a \cdot t_b \leq 0$, therefore

$$\begin{aligned} p(\mathbf{x}, \varphi|\mathbf{a}, \mathbf{b}) = & \int_{-\infty}^{0} p_n(\mathbf{x} + t\Delta_\varphi; \mathbf{a}, \sigma^2)dt \int_{0}^{\infty} p_n(\mathbf{x} + t\Delta_\varphi; \mathbf{b}, \sigma^2)dt \\ & + \int_{0}^{\infty} p_n(\mathbf{x} + t\Delta_\varphi; \mathbf{a}, \sigma^2)dt \int_{-\infty}^{0} p_n(\mathbf{x} + t\Delta_\varphi; \mathbf{b}, \sigma^2)dt \end{aligned} \tag{2.4}$$

According to Lemma 2.1, Equation A.8 is equivalent to Equation 2.3. □

Proposition 2.1. *Let ℓ' be a line segment transformed from line segment ℓ in 2D space by nonsingular 3×3 projection matrix $\mathbf{P}$. If the two ending points of ℓ are random variables drawn from normal distributions $N(\mathbf{a}, \sigma^2)$ and $N(\mathbf{b}, \sigma^2)$ respectively, then for any $\mathbf{x}$, the probability that $\mathbf{x}$ lies on ℓ' and ℓ' has slope angle φ is*

$$p_{proj}(\mathbf{x}, \varphi|\mathbf{P}, \mathbf{a}, \mathbf{b}) = p((x', \varphi') = proj(\mathbf{P}^{-1}, \mathbf{x}, \varphi)|\mathbf{a}, \mathbf{b}) \tag{2.5}$$

where proj$(\mathbf{Q}, \mathbf{x}, \varphi)$ *is a function returns the corresponding coordinate and slope angle with respect to* $\mathbf{x}$ *and* φ *after projection transformation* $\mathbf{Q}$.

Proof. The mapping from $(\mathbf{x}, \varphi)$ to $(\mathbf{x}', \varphi')$ is bijective, thus Equation 2.5 holds. □

The point coordinate transformed by $\mathbf{Q}$ can be obtained by homogeneous coordinate representation. For the slope angle, let $\mathbf{q}_i$ be the i-th row vector of projection matrix $\mathbf{Q}$, the transformed slope angle φ' at location $\mathbf{x} = (x, y)^\top$ is

$$\varphi' = \arctan \frac{f(\mathbf{q}_2, \mathbf{q}_3, x, y, \varphi)}{f(\mathbf{q}_1, \mathbf{q}_3, x, y, \varphi)} \tag{2.6}$$

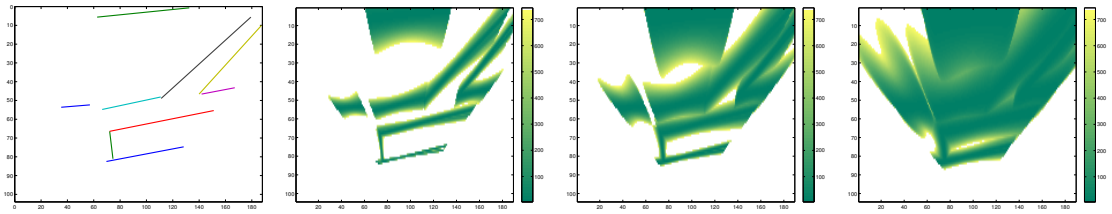
where

$$\begin{aligned} f(\mathbf{u}, \mathbf{v}, x, y, \varphi) =& (u_2 v_1 - u_1 v_2)(x \sin\varphi - y \cos\varphi) \\ &+ (u_1 v_3 - u_3 v_1) \cos\varphi + (u_2 v_3 - u_3 v_2) \sin\varphi \ . \end{aligned} \tag{2.7}$$

According to the above, for each pixel location in the recovered view of a ground image, the probability that the pixel lies on a line segment given a slope angle can be computed in closed form. Figure 2.6 shows an example probability distribution for line segments under uncertainty. It can be observed from the plot that more uncertainty is associated with line segments farther from the camera and is resulted from a larger σ value.

2.6 Geometric matching under uncertainty

Our approach to planar structure matching is motivated by chamfer matching. Chamfer matching efficiently measure the similarity between two sets of image



(a) (b) $\sigma = 0.5$ (c) $\sigma = 1$ (d) $\sigma = 2$

Figure 2.6: Examples of uncertainty modeling: (a) the ortho-rectified line segments (b-d) the negation of probability log map for points on lines. The probability for each pixel location is obtained by summing up the probabilities for all discretized orientations. The camera is located in the image center and faces upward.

features by evaluating the sum of distances between each feature in one image and its nearest feature in the other image [60]. More formally,

$$D_c(\mathbf{A}, \mathbf{B}) = \sum_{\mathbf{a} \in \mathbf{A}} d(\mathbf{a}, \arg\min_{\mathbf{b} \in \mathbf{B}} d(\mathbf{a}, \mathbf{b})) \tag{2.8}$$

where $\mathbf{A}, \mathbf{B}$ are two sets of features, and $d(\cdot, \cdot)$ is the distance measure for a feature pair. Commonly, feature sets contain only the 2D coordinates of points, even if those points are sampled from lines that also have an associated orientation. Oriented chamfer matching (OCM) [61] makes use of point orientation by modifying the distance measure to include the sum of angle differences between each feature point and its closest point in the other image. Another way to incorporate orientation is directional chamfer matching (DCM) [45] which defines features to be, more generally, points in 3D space (x-y coordinates and orientation angle). This approach uses the same distance function as the original chamfer matching but has a modified feature distance measure. We follow the DCM method [45] to define our feature

space. In our case, point orientation is set to the slope angle of the line it lies on.

2.6.1 Notation

All of the points in our formulation are in the 3D space. A point feature is defined as $\mathbf{u} = (\mathbf{u}_l, u_\phi)$ where $\mathbf{u}_l$ represents the 2D coordinates in real world and u_ϕ is the orientation associated with location $\mathbf{u}_l$. $\mathbf{G}_p$ is the set of points $\{\mathbf{g}\}$ in the ground image with uncertainty modeled by probability distribution $p(\cdot)$. $\mathbf{O}$ is the set of points in the ortho image. $\mathbf{L}_G$ is the set of annotated line segments in the ground image. A line segment is defined as $\boldsymbol{\ell} = (\mathbf{a}_\ell, \mathbf{b}_\ell)$ where $\mathbf{a}_\ell$ and $\mathbf{b}_\ell$ are the end points of $\boldsymbol{\ell}$. For any line segment $\boldsymbol{\ell}$ and an abitrary line segment $\hat{\boldsymbol{\ell}}$ in the feature space, $p(\hat{\boldsymbol{\ell}}|\boldsymbol{\ell})$ is the confidence of $\hat{\boldsymbol{\ell}}$ by observing $\boldsymbol{\ell}$.

2.6.2 Distance metric

The feature distance for $\mathbf{u}, \mathbf{v}$ is defined as

$$d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_g = \|\mathbf{u}_l - \mathbf{v}_l\|_2 + |u_\phi - v_\phi|_a \tag{2.9}$$

where $\|\mathbf{u}_l - \mathbf{v}_l\|_2$ is the Euclidean distance between 2D coordinates in meters and $|u_\phi - v_\phi|_a = \lambda \min(|u_\phi - v_\phi|, \pi - |u_\phi - v_\phi|)$ is the smallest difference between two angles in radians. The parameter λ relates the unit of angle to the unit of world distance. We choose $\lambda = 1$ so that π angle difference is equivalent to around 3.14 meters in the real world. For this feature space definition, the chamfer distance in Equation 2.8 can be efficiently computed by pre-computing the distance transform for the reference image (refer to [45, 46] for more details) and convolving the query

image with the reference distance transform.

2.6.3 Formulation

The distance function for matching ground image $\mathbf{G}_p$ to ortho image $\mathbf{O}$ is formulated as

$$D(\mathbf{G}_p, \mathbf{O}) = D_m(\mathbf{G}_p, \mathbf{O}) + D_\times(\mathbf{G}_p, \mathbf{O}) \tag{2.10}$$

where D_m is the probablistic chamfer matching distance and $D_\times$ is a term penalizing line segment crossings. The probablistic chamfer matching distance is defined as

$$D_m(\mathbf{G}_p, \mathbf{O}) = \frac{1}{|\mathbf{L}_G|} \sum_{\boldsymbol{\ell} \in \mathbf{L}_G} \int p(\hat{\boldsymbol{\ell}}|\boldsymbol{\ell}) \int p(\mathbf{g}|\hat{\boldsymbol{\ell}}) \left(\min_{\mathbf{o} \in \mathbf{O}} \|\mathbf{g} - \mathbf{o}\|_g \right) d\mathbf{g} d\hat{\boldsymbol{\ell}} \ . \tag{2.11}$$

The marginal distribution $\int p(\hat{\boldsymbol{\ell}}|\boldsymbol{\ell}) p(\mathbf{g}|\hat{\boldsymbol{\ell}}) d\hat{\boldsymbol{\ell}} = p(\mathbf{g}|\boldsymbol{\ell})$ is the probability that point $\mathbf{g}_l$ lies on line segment $\boldsymbol{\ell}$ with slope angle g_ϕ. Equation 2.11 is equivalent to

$$D_m(\mathbf{G}_p, \mathbf{O}) = \frac{1}{|\mathbf{L}_G|} \sum_{\boldsymbol{\ell} \in \mathbf{L}_G} \int p(\mathbf{g}|\boldsymbol{\ell}) \left(\min_{\mathbf{o} \in \mathbf{O}} \|\mathbf{g} - \mathbf{o}\|_g \right) d\mathbf{g} \tag{2.12}$$

whose discrete representation is

$$D_m(\mathbf{G}_p, \mathbf{O}) = \sum_{\mathbf{g}} p'(\mathbf{g}|\mathbf{L}_G) \left(\min_{\mathbf{o} \in \mathbf{O}} \|\mathbf{g} - \mathbf{o}\|_g \right) \tag{2.13}$$

where $p'(\mathbf{g}|\mathbf{L}_G) = \frac{1}{|\mathbf{L}_G|} \sum_{\boldsymbol{\ell} \in \mathbf{L}_G} \frac{p(\mathbf{g}|\boldsymbol{\ell})}{\sum_{\mathbf{g}} p(\mathbf{g}|\boldsymbol{\ell})}$ is the probability of points lying on the structure and each line segment equally contributes to the distance value. In fact, Equation 2.12 is equivalent to the original chamfer matching (Equation 2.8) if no uncertainty is present.

Intersections between ortho line segments and ground line segments indicate low matching quality. Therefore, we add an additional term into our formulation to

penalize camera poses that result in too many line segment intersections. The cross penalty for line segments is defined as

$$D_{\times}(\mathbf{G}_p, \mathbf{O}) = \frac{\sum_{\boldsymbol{\ell}\in\mathbf{L}_G} \int p(\hat{\boldsymbol{\ell}}|\boldsymbol{\ell}) \sum_{\mathbf{o}\in\mathbf{O}} \int p(\mathbf{g}|\hat{\boldsymbol{\ell}}) |g_\phi - o_\phi|_a \delta(\mathbf{g}_l - \mathbf{o}_l) d\mathbf{g} d\hat{\boldsymbol{\ell}}}{\sum_{\boldsymbol{\ell}\in\mathbf{L}_G} \int p(\hat{\boldsymbol{\ell}}|\boldsymbol{\ell}) \sum_{\mathbf{o}\in\mathbf{O}} \int p(\mathbf{g}|\hat{\boldsymbol{\ell}}) \delta(\mathbf{g}_l - \mathbf{o}_l) d\mathbf{g} d\hat{\boldsymbol{\ell}}} \tag{2.14}$$

where $\delta(\cdot)$ is the delta function. This function is is a normalized summation of angle differences for all intersection locations, which are point-wise equally weighted. Because $\int p(\hat{\boldsymbol{\ell}}|\boldsymbol{\ell}) p(\mathbf{g}|\hat{\boldsymbol{\ell}}) d\hat{\boldsymbol{\ell}} = p(\mathbf{g}|\boldsymbol{\ell})$, the function is equivalent to

$$D_{\times}(\mathbf{G}_p, \mathbf{O}) = \frac{\sum_{\boldsymbol{\ell}\in\mathbf{L}_G} \int p(\mathbf{g}|\boldsymbol{\ell}) \sum_{\mathbf{o}\in\mathbf{O}} |g_\phi - o_\phi|_a \delta(\mathbf{g}_l - \mathbf{o}_l) d\mathbf{g}}{\sum_{\boldsymbol{\ell}\in\mathbf{L}_G} \int p(\mathbf{g}|\boldsymbol{\ell}) \sum_{\mathbf{o}\in\mathbf{O}} \delta(\mathbf{g}_l - \mathbf{o}_l) d\mathbf{g}} \tag{2.15}$$

whose equivalent discrete formulation is

$$D_{\times}(\mathbf{G}_p, \mathbf{O}) = \frac{\sum_{\mathbf{g}} p'(\mathbf{g}|\mathbf{L}_G) \sum_{\mathbf{o}\in\mathbf{O}} |g_\phi - o_\phi|_a \delta[\mathbf{g}_l - \mathbf{o}_l]}{\sum_{\mathbf{g}} p'(\mathbf{g}|\mathbf{L}_G) \sum_{\mathbf{o}\in\mathbf{O}} \delta[\mathbf{g}_l - \mathbf{o}_l]} \tag{2.16}$$

where $p'(\mathbf{g}|\mathbf{L}_G)$ is defined in Equation 2.6.3 and $\delta[\cdot]$ is the discrete delta function.

2.6.4 Hypothesis generation

Given a ground image $\mathbf{G}_p$, the score for ortho image $\mathbf{O}_i$ corresponds to one of the candidate geolocations. is evaluated as the minumum possible distance, so the estimated fine camera pose within ortho image $\mathbf{O}_i$ is

$$\hat{\mathbf{x}}_i = \hat{\mathbf{x}}(\mathbf{O}_i, \mathbf{G}_p) = \arg\min_{\mathbf{x}_l, x_\phi} D(\mathbf{R}_{x_\phi}\mathbf{G}_p + \mathbf{x}_l, \mathbf{O}_i) \tag{2.17}$$

where $\mathbf{R}_\alpha$ is the rotation matrix corresponded to angle α.

2.6.5 Implementation remark

The two distance functions can be computed efficiently based on distance transforms in which the orientations are projected into 60 uniformly sampled angles

and the location of each point is at the pixel level. Firstly, probability $p(\mathbf{g}|\boldsymbol{\ell})$ can be computed in closed form according to Proposition 2.1. So the distribution $p'(\mathbf{g}|\mathbf{L}_G)$ can be pre-computed for each ground image. Based on 3D distance transform [45], Equation 2.13 can be computed with a single convolution operation. The computation of Equation 2.16 involves delta functions, which is essentially equivalent to a binary indicator mask for an ortho image: $M_{\mathbf{O}}(\mathbf{x}) = 1$ means there exists a point $\mathbf{o} \in \mathbf{O}$ located at coordinate $\mathbf{x}$ and 0 means there is no feature at this position. Such indicator mask can be directly obtained. So we compute for every orientation φ and location $\mathbf{x}$ a distance transform $A_\varphi(\mathbf{x}) = \sum_{\mathbf{o}\in\mathbf{O}\wedge\mathbf{o}_l=\mathbf{x}} |\varphi - o_\phi|_a$. The denominator of Equation 2.16 can be computed directly by convolution, while the numerator needs to be computed independently for each orientation. For a discretized orientation θ, a matrix is defined $W(\mathbf{g}) = p'(\mathbf{g}|\mathbf{L}_G)M_{\mathbf{O}}(\mathbf{g}_l)$ for all $\mathbf{g}$ such that $g_\phi = \theta$ and otherwise $W(\mathbf{g}) = 0$. Convolving matrix W with the distance transform A_θ will achieve partial summation of Equation 2.16. Summing them up for all orientations gives the numerator in Equation 2.16.

2.7 Experiment

2.7.1 Dataset

We build a data set from Google Maps with an area of around $1km \times 1km$. 35 ground images are randomly extracted from Google Street View together with their ground truth locations. Each ground image is a 640×640 color image. Field of view information is retrieved. A total of 400 satellite images are extracted using a sliding

Figure 2.7: Example ground images (upper) and ortho images (lower) from our dataset. The ground image can be taken anywhere within one of the satellite images. ©Google

window within this area. Each ortho photo is also a 640×640 color image. The scale of ortho images is 0.1 meters per pixel. 10 ground images are used for experiments on the uncertainty parameter σ and the remaining 25 ground images are used for testing. Example ground and satellite images are shown in Figure 2.7. Geolocation in this dataset is challenging because most of the area share highly similar visual appearance.

2.7.2 Evaluation criterion

Three quantitative criteria are employed to evaluate the experiments. First, we follow previous work [42] by using curves on *percentage of ranked candidate* vs. *percentage of correctly localized images*. By ranking all the ortho images in

descending order of their matching scores, *percentage of ranked candidates* is the percentage of top ranked images in all of the ortho images and *percentage of correctly located images* is the percentage of all the queries whose ground truth locations are among the corresponding top ranked candidate images. Second, we obtain a overall score by counting the area under this curve (AUC). A higher overall score generally means more robustness in the algorithm. Third, we look into the *percentage of correctly localized images* among 1%, 2%, 5% and 10% top ranked locations.

2.7.3 Parameter selection

Intuitively, σ represents the pixelwise variance of the line segment end points, so it should not be more than several pixels. We randomly pick 10 ground images and 20 ortho images including all ground-truth locations to compose training set for tuning σ. The geolocation performance over a set of σ values ranged from 0 to 3 with a step 0.5 are evaluated and shown in Figure 2.8 where $\sigma = 0$ means no uncertainty model is used. The peak is reached when the σ is between 1.5 and 2. Therefore, we fix $\sigma = 2$ in all of the following experiment.

2.7.4 Results

Our geometric matching approach returns distance values densely cover every pixel and each of the 12 sampled orientations in each ortho image. The minimum distance is picked as the score of an ortho image. Therefore, our approach not only produces ranking among hundreds of ortho images but also shows possible camera

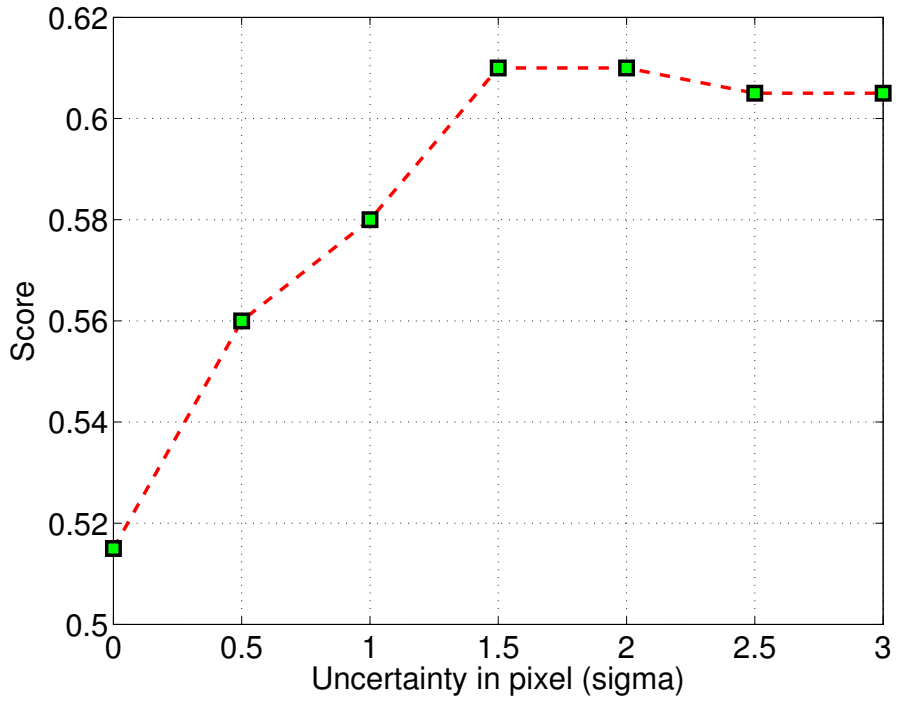


Figure 2.8: Geolocation AUC score under different uncertainty variances σ where $\sigma = 0$ represents the approach without uncertainty modeling.

Table 2.1: Comparison among oriented chamfer matching [61], directional chamfer matching [45] and our approach. The uncertainty model is evaluated for each method. For each evaluation criterion, the highest score is highlighted in **bold** and the second one highlighted with underline. Our uncertainty based formulation is top among all these methods. Both of the three methods can be improved by our uncertainty model. OCM boosts its performance when incorporated with our probablistic representation.

	w/o uncertainty			**w/ uncertainty**		
Method	OCM	DCM	ours	OCM	DCM	ours
Top 1%	0.08	0.00	0.00	0.04	0.00	**0.12**
Top 2%	0.08	0.04	0.08	0.04	0.04	**0.20**
Top 5%	0.16	0.12	0.12	0.20	0.12	**0.32**
Top 10%	0.24	0.24	0.28	0.28	0.28	**0.44**
Score(AUC)	0.6814	0.7419	0.7500	0.7688	0.7577	**0.8219**

locations and orientations.

We compare our approach with two existing matching methods i.e. oriented chamfer matching [61] and directional chamfer matching [45]. To study the effectiveness of our uncertainty models, we also evaluate these methods with uncertainty model embedded. DCM is equivalent to the first term D_m in our formulation. OCM is to find the nearest feature in the other image and compute the sum of pixel-wise distance and the angle differences to the same pixel. We apply our uncertainty model into their formulation in a similar way as the probablistic chamfer matching distance does. Thus, in total we have six approaches in our comparison. Their performance curves are shown in Figure 2.9. Over 90% of the ground queries can be correctly located when half of the ortho images are rejected. Numerical results are in Table 2.1. While our approach significantly outperforms at any percentage of retrieved images, our performance improvement is particularly large for top ranked images.

Four successfully localized queries are shown in Figure 2.10. For these ground images, the ground truth locations are included in the top 5 ranked candidate ortho images out of 400. From this visualization, few labeling errors can be noticed from miss-alignment between ortho images and rectified line segments. Among these top responses, most false alarms are building roofs. A common property is that they have relatively denser line features. Another issue is the line detection in ortho images does not handle shadows well. Most linear structures in these shadow areas are not detected.

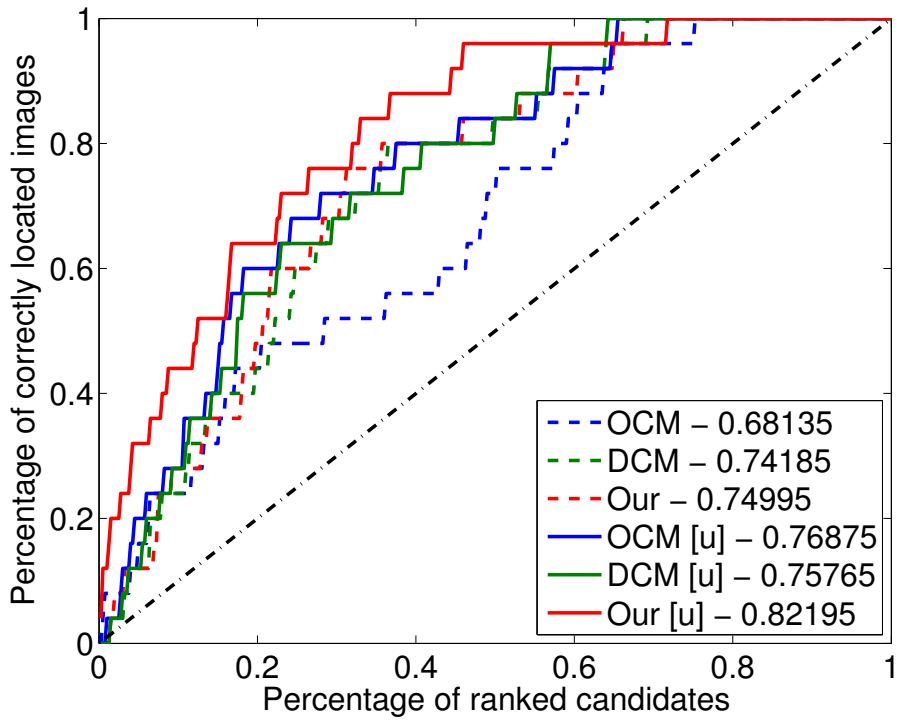


Figure 2.9: Performance curve for six approaches: the ortho images are ranked in ascending order. The x-axis is the number of selected top ranked ortho images and the y-axis is the total number of ground image queries whose true locations are among these selected ortho images. The overall AUC scores are shown in the legend where "[u]" means "with uncertainty modeling". The black dash-dot line indicates chance performance.

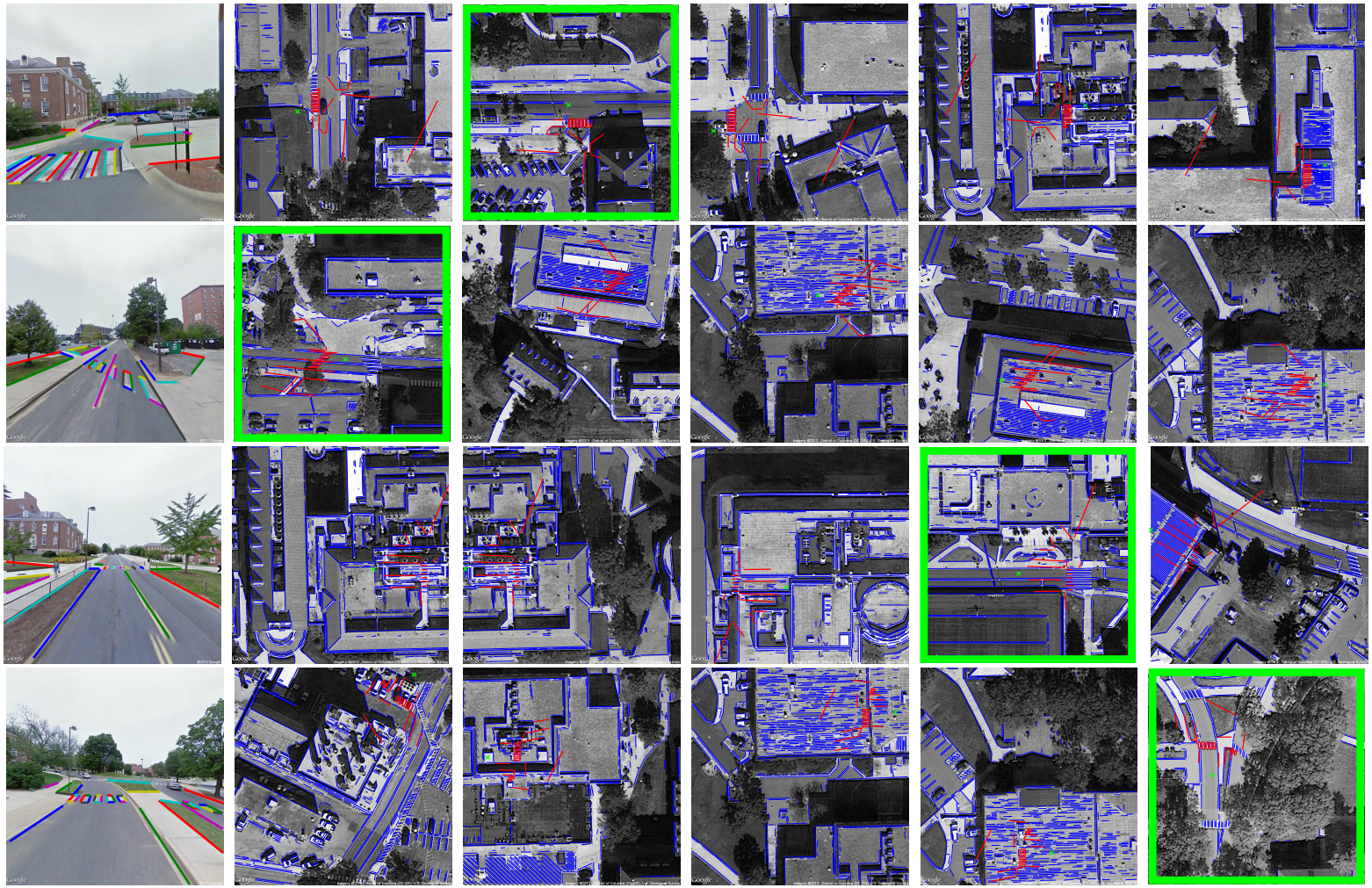


Figure 2.10: Four queries successfully geolocated within top five candidates are shown. The leftmost column is the ground image with annotated line segments. For each query, top five scoring ortho images are shown in ascending order of their rank. Ground-truths are highlighted by green bounding boxes. For each ortho image, blue lines are automatically detected and red lines are parsed from ortho-rectified ground images. Green cross indicates the most probable camera location within that ortho image.

2.8 Discussions and future work

2.8.1 Automatic line segment annotation

The cross-view image matching problem has been one of the most challenging problems in computer vision. Our approach essentially finds a common set of linear geometric features that is visible from both viewpoints. Although the linear structure preserves the invariance of a visual scene and tends to be much more reliable than color or intensity based features, it is still observed that the process of extracting such line segment features could be greatly improved. Multiple future directions could be taken in this area such as automating the ground line segment detection and horizon line detection in the ground-based photos, and removing noisy linear features that are not lying on the ground in the ortho photos.

2.8.2 Joint and iterative matching

The proposed feature matching is based on an one-time estimation of the projective transformation, which can be noisy. Another interesting direction is to explore how the parameters in projection matrices could be jointly estimated together with the feature matching. This would be a challenging problem in terms of efficiency, considering the large scale of the satellite imagery.

2.8.3 Invariant feature learning

From the perspective of feature learning, one natural question to ask is whether it is possible to learn directly a feature representation that can be used for visual reasoning under unknown projective transformations. And it would be interesting to see if we could learn such representation in an unsupervised or weakly supervised way, considering the lack of registered (or corresponded) training image pairs.

2.9 Conclusion

We investigated the single image geolocation problem by matching human annotated line segments in the ground image to automatically detected lines in the ortho images. An uncertainty model is devised for line segments under projective transformations. Using this uncertainty model, ortho-rectified ground images are matched to candidate ortho images by distance transform based methods. The experiment has shown the effectiveness of our approach in geographic areas with similar local appearances.

Chapter 3: Spatially robust encoding of low-level visual features

3.1 Motivation

As face recognition techniques have gradually matured over the past few decades, the research focus has shifted from recognizing faces with controlled variations to unconstrained real-world scenarios [66]. Modern approaches based on high dimensional feature encoding [67–70] and deep neural networks [71, 72] have recently emerged and achieved promising results on unconstrained face databases [73, 74]. However, most existing face recognition systems depend on accurate face detection and registration. Unfortunately, these two components are a significant source of error in real-world environments or real-time applications.

In the application of mobile face authentication, for example, faces recorded from a front-facing smartphone camera often exhibit rare non-horizontal poses (i.e., neither frontal nor profile) and are often partly outside the camera's viewpoint. This problem is exacerbated when users are performing other tasks (as opposed to actively ensuring that their face is within the camera view) in which case the facial video quality becomes even worse, further challenging existing face detection and registration systems. For example, one of our experiments shows that the popular Viola-Jones face detector [75] fails on a significant portion of a smartphone-recorded

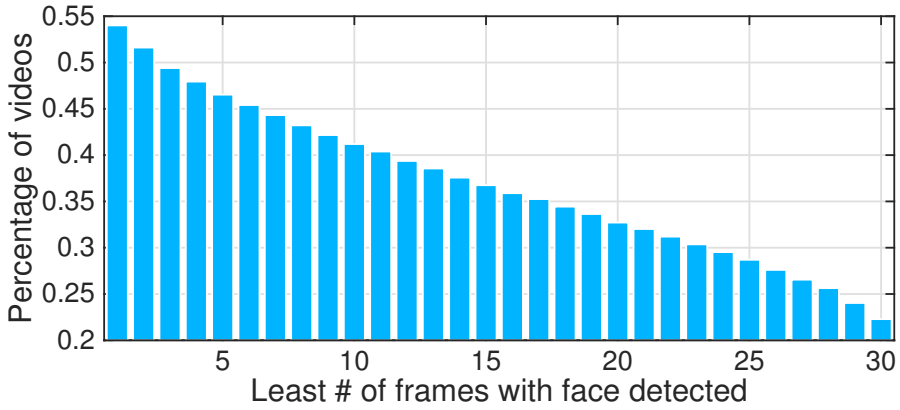


Figure 3.1: Performance of Viola-Jones (OpenCV) multi-scale face detector on a mobile based video face authentication dataset [76] with a total of 19,158 sampled video clips each 30 frames long. The x-axis is the number of frames in each video and the y-axis shows the percentage of video clips with at least the corresponding number of frames having faces detected. While all of the video clips contain faces, only 54% of the videos have at least one face detected and 22% have faces detected across all the 30 frames.

face dataset [76] (Figure 3.1).

Most current face recognition datasets use images viewed from a distance for benchmarking. This type of data involves other challenges, compared to those from mobile applications: low image resolution and background distractions, because of which we can still expect some degree of errors in the detection step, i.e., improper estimation of face centers and bounding box sizes. A statistical illustration of the face detection errors using FDDB benchmark data [77] is shown in Figure 3.2.

Motivated by these observations, we explore the possibility of addressing unconstrained face verification problems without explicit face detection or registration. The central idea of our approach is that the codebook can be optimized to encode additional information for discriminating relevant image patches from irrel-

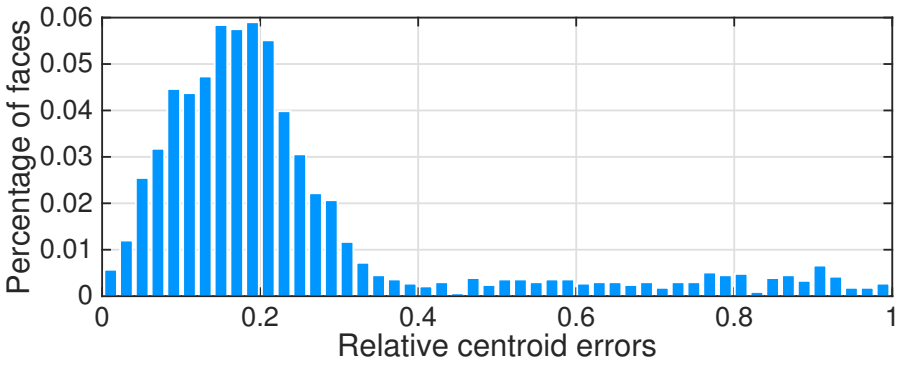


Figure 3.2: Viola-Jones (OpenCV) multi scale face detection results on Face Detection Dataset and Benchmark (FDDB) [77]: the relative centroid errors are computed as the centroid distance between detected faces and their closest ground truths, divided by the averaged axis length of ground truth ellipses. The chart shows 68% of faces are detected faces while the other 32% are false alarms with no overlap with any of the ground truth faces. Notably, 50% of the detected faces produce some levels of offsets from 0% to 25% where the peak is around 20% of the face size (e.g., for 150×150 faces, the peak of errors is 30 pixels).

evant background distractions. We propose a unified codebook-based framework, named "selective encoding", the core of which is a component named "selector" which injects trained relevance information into codewords via a set of "relevance weights" and utilizes these weights to select semantically relevant patch descriptors and codewords at test time. Patch descriptors and codewords that successfully pass the selector will be used for encoding images. The selector essentially finds a good relevant sub-matrix of the posterior probability (assignment) matrix for feature encoding.

For recognizing unreliably localized faces, we define the descriptor relevance as foreground probabilities, so image patches belonging to the facial region are selected over those that do not. The relevance distribution training involves counting for each

codeword the foreground/background distribution of its assigned patch descriptors. These distributions are used for computing the foreground probability of each newly observed patch in testing. Background distractions are thereafter removed from the descriptor set so that the encoded representation can achieve spatial robustness.

Fisher vector encoding [78] is one of the most powerful codebook based feature encoding techniques. However, its most recent applications in face verification require face detection and registration. One of our experiments shows that this method degrades quickly with inaccurate estimation of face centers and bounding box sizes due to the inclusion of more distractive patches. We validate our framework using the Fisher vector encoding on public datasets and show that our method is capable of robustifying such encoding technique with respect to uncertain face localization. We further apply our framework to a mobile based active face authentication task to show its applicability in real-world scenarios.

Contribution. The main contributions of this work include

- A generic and unified framework for selecting and encoding relevant features which does not require accurate detection or registration;
- The application of Selective encoding to Fisher vector encoding for spatially robust face verification;
- The application of Selective Fisher vector encoding to mobile based active face authentication.

3.2 Related work

3.2.1 Feature encoding

The bag of visual words model [23] is the most popular feature encoding framework for many computer vision tasks. In this model, a codebook is built using K-means clustering and each feature is assigned a weight for each cluster center (aka. codeword) according to their distances. An image is thereafter represented by the distribution (histogram) of those assignments. Most modern feature encoding techniques are extensions of this codebook model such as Fisher vectors [35] and the vector of locally aggregated descriptors [34]. The central idea is that, instead of using only an assignment distribution, an image can also be represented using the first order (mean of difference) and second order (standard deviation) statistics of all the (soft or hard) assigned features for each codeword. Fisher vector encoding is now among the state-of-the-art on various computer vision applications such as image classification [35, 69, 78], image retrieval [79] and face verification [68]. Our work is built upon Fisher vectors and integrates additional supervised information into the codebook for encoding semantically relevant patches.

3.2.2 Unconstrained face recognition

The upsurge of research on unconstrained face recognition gave rise to the creation of Labeled faces in the wild (LFW) dataset [73]. Besides the Fisher vector faces [70], many works have been developed on this topic, such as high dimensional

local binary patterns [67], deep learning based approaches [71,72] and sparse coding based approaches [80, 81]. Considering that face recognition problems are often challenged by pose variations, many works try to improve the recognition accuracy by means of robust facial alignment and correction using sophisticated 3D models or shape matching [66, 72, 80, 82]. However, the vulnerablility of face detectors under real-world scenarios is usually overlooked and most existing face verification methods generally assume detected and well aligned faces are given [68, 70]. The goal of our work is to remove the strong dependency on face detection by improving the encoding scheme to be significantly more robust to spatial misalignments.

3.2.3 Joint localization and classification

The general image object classification problems are also affected by the performance of object localization. Most works try to find good localization and segmentation of the objects to relieve the subsequent recognition task [83,84]. However, detection is even harder than classification in some sense (e.g., robust bounding box estimation). A few recent works are motivated by the idea of jointly detecting and classifying objects in images in the hope that the two tasks help each other. Nguyen et al. [85] proposed to jointly localizing discriminative regions and training a region-based SVM for image categorization. Lan et al. [86] proposed a figure-centric model learned by latent SVM for joint action localization and recognition. The most similar work to ours is the object-centric pooling [87]. Its main idea is to infer, jointly with classification, tight object bounding boxes and pool features within detected

regions. They developed an MIL-like SVM formulation for joint object localization and classification. However, our work differs in that (1) instead of finding perfect detections, we explore the implicit feature selection power of the codebook, and (2) our framework is designed for feature encoding and does not depend on any subsequent classification.

3.3 Preliminary – Fisher vector encoding

The Fisher vector (FV) encoding was first proposed in [35] and applied to face verification problems in [70] and [68]. The central idea of Fisher vector encoding is to aggregate higher order statistics of each codebook into a high dimensional feature. More specifically, a Gaussian mixture model (GMM) is trained as the visual codebook. First-order and second-order distance statistics w.r.t.each of the Gaussian mixture components are concatenated into the final feature representation. Let $\mathbf{x}_p$ be the p-th descriptor and $(\mu_k, \sigma_k^2, \pi_k)$ be the k-th Gaussian component. The assignment coefficient (posterior probabilities) of $\mathbf{x}_p$ with respect to the k-th Gaussian is represented using $\alpha_k(\mathbf{x}_p)$. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ be the descriptor set, the Fisher vector representation is computed as $\phi(\mathbf{X}) = [\Phi_1^{(1)}, \Phi_1^{(2)}, \ldots, \Phi_K^{(1)}, \Phi_K^{(2)}]$ where

$$\Phi_{ik}^{(1)} = \frac{1}{N\sqrt{\pi_k}} \sum_{p=1}^{N} \alpha_k(\mathbf{x}_p) \left(\frac{x_{ip} - \mu_{ik}}{\sigma_{ik}} \right) , \tag{3.1}$$

$$\Phi_{ik}^{(2)} = \frac{1}{N\sqrt{2\pi_k}} \sum_{p=1}^{N} \alpha_k(\mathbf{x}_p) \left[\left(\frac{x_{ip} - \mu_{ik}}{\sigma_{ik}} \right)^2 - 1 \right] . \tag{3.2}$$

Most existing works on Fisher vectors apply signed square root and ℓ^2 normalization to the feature vectors which tend to further improve the representation capability of Fisher vectors [69, 70].

3.4 Selective encoding overview

The proposed selective encoding framework is illustrated in Figure 3.3. Existing codebook based face recognition approaches require detection and registration beforehand, while our framework reduces the need for such prerequisites. Generally speaking, our framework is composed of three main stages: (1) building a vocabulary (2) descriptor and codeword selection (selector) and (3) feature encoding. The key component for achieving spatial robustness is the selector, which selects relevant descriptors and codewords for the feature encoding stage. The selector is trained with weakly supervised prior knowledge on the descriptor relevance (i.e., rough detection bounding boxes). An advantage of our framework is that we do not require any extra computational cost during testing because the selector is essentially performed on the matrix of posterior probabilities (assignment) for the codebook, which is necessarily computed in the conventional codebook framework.

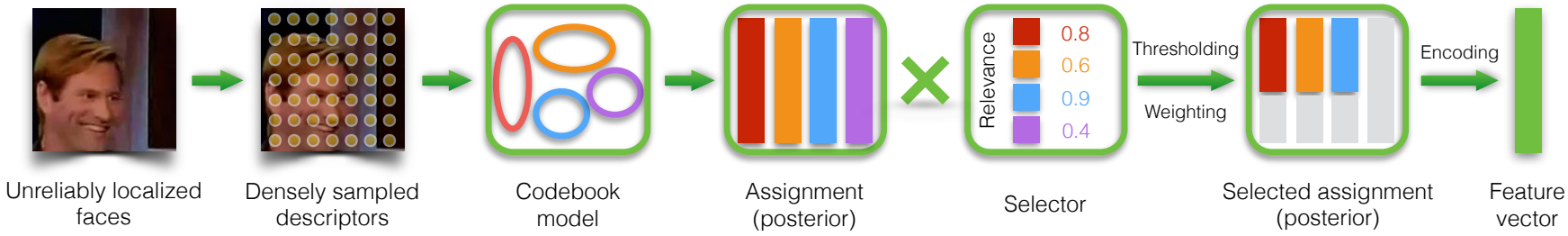


Figure 3.3: The proposed selective encoding framework: Images or videos with unreliably localized faces are the direct input to our model. Posterior probabilities (assignment) for densely sampled local descriptors are computed according to the trained codebook model. The relevance weight of each descriptor is calculated according to its posterior probability distribution and the relevance of corresponding codewords. The selector component is trained offline using weakly supervised features. A subset of the assignment matrix (or a new assignment matrix) is generated by thresholding (or re-weighting using) the descriptor relevance, and used for image feature encoding.

3.5 Vocabulary

3.5.1 Descriptor extraction

Following [70], we extract densely sampled SIFT descriptors [88] at 5 different scales. The 128-D descriptors are further reduced to 64-D by principal component analysis. Fisher vectors are often learned using an augmented descriptor which adds two additional dimensions for the spatial coordinates of each SIFT descriptor. A normalization is utilized for the augmented dimension, i.e., $[x_{\text{aug}}, y_{\text{aug}}] = [\frac{x}{w} - 0.5, \frac{y}{h} - 0.5]$ where w, h are the width and height of the window.

3.5.2 Codebook construction

The Fisher vector encoding uses Gaussian mixture models to provide softer structures and capture smoother feature distributions in the encoding than the K-means clustering based codebook. As [70], we use 512 Gaussian components for our experiments.

3.6 Selector

The selector consists of two parts: (1) descriptor selection and (2) codeword selection. Both stages are executed based on the trained relevance weights of each codeword and their corresponding posterior probabilities w.r.t.newly observed image patches.

3.6.1 Codeword relevance

Given a trained codebook (Gaussian mixture model), the selector is trained to associate additional foreground/background information with each codeword (Gaussian component). The training involves calculation of the relevance weights for each codeword.

Let $\mathbf{x}_i$ be the i-th patch descriptor, $\boldsymbol{\theta}_k$ be the k-th Gaussian mixture component and their corresponding posterior probability be $p(\boldsymbol{\theta}_k|\mathbf{x}_i)$. The selector is trained using n-dimensional patch descriptors $\mathbf{x}_i \in \mathbb{R}^n$ with their binary labels $y_i \in \{0, 1\}$ which represent whether they should be selected for feature encoding, by counting for each codeword the expected descriptor relevance, i.e.,

$$p_s^c(\boldsymbol{\theta}_k) = \frac{\sum_{i=1}^{N} p(\boldsymbol{\theta}_k|\mathbf{x}_i) y_i}{\sum_{i=1}^{N} p(\boldsymbol{\theta}_k|\mathbf{x}_i)} \ . \tag{3.3}$$

The codeword relevance value ranges between 0 and 1. Codewords with higher relevance weights (larger than 0.5) are more likely to aggregate foreground descriptors while those with lower relevance weights (lower than 0.5) have higher chance of being background. Although keeping unnecessary codewords will not damage the encoding space, discarding those background codewords naturally reduces the feature dimension and in some cases improves the recognition accuracy (Figure 3.11(b)).

For recognizing unregistered faces, the training patches and their semantic labels are obtained by using images with valid detection outputs. Those features located within detected face bounding boxes are labeled as 1 and those outside labeled as 0. In our experiments we are using loose detection bounding boxes which

contain background areas; however, the learned relevance distributions is sufficient for improving the encoding robustness.

3.6.2 Descriptor relevance

At test time, the posterior probabilities for each patch descriptor are given from the codebook model. The descriptor relevance weight is then computed by counting the relevance contribution from each codeword with respect to their posterior probabilities, i.e.,

$$p_s^d(\mathbf{x}_i) = \sum_{k=1}^{K} p(\boldsymbol{\theta}_k|\mathbf{x}_i) p_s^c(\boldsymbol{\theta}_k) \ . \tag{3.4}$$

The posterior probability can be computed via either soft or hard assignment (in hard assignment settings, the highest posterior probability for each descriptor is lifted to 1 and all the others reduced to 0). The descriptor relevance also ranges between 0 and 1, similar to codeword relevance. Intuitively, the descriptor selection plays a key role in achieving spatial robustness of feature encoding by removing background patches. In our experiment, we remove all descriptors with relevance lower than 0.5 (a threshold for separating foreground from background) for patch selection.

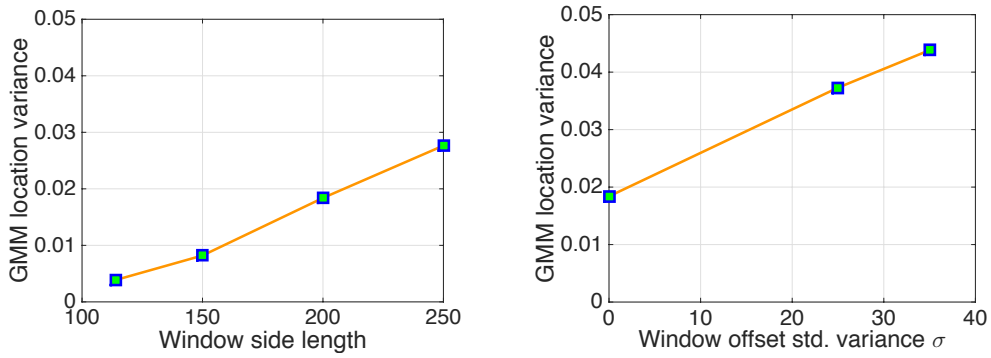
3.7 Encoding

The encoding stage receives from the selector a subset (or a modified version) of the posterior probability matrices and encodes them as Fisher vectors (as described in Section 3.3). The encoded Fisher vectors can be further reweighed or reduced to

lower dimensions by multiple metric learning approaches; however, with restricted training samples, learning a low rank metric is difficult [70]. The mobile face authentication problem comes with a limited training set – users are not likely to spend much time actively training the smartphones. So in our experiments, we employ the ℓ^2 metric and diagonal metric learning (i.e., training a diagonal metric using support vector machines) proposed in [70] for evaluating encoding performance.

3.8 Learning with spatial-sensitive features

Intuitively, the location features help when the face images are properly registered. However, when the registration is poor, augmented location information may instead hurt the performance. The GMM model can smooth out the Gaussian component on the location dimensions (Figure 3.4) and may also learn the location distribution of patches when the training images have some underlying mis-registration patterns. However, the robustness to localization errors is not sufficient for unconstrained spatial patterns, in which case performance drops quickly and becomes worse than ignoring location information altogether. The main reason is because patches belong to the same facial part are assigned to different codewords due to the influence of the augmented location dimension. However, our framework can adapt to such location sensitive augmented features. The central idea is that we can identify relevant patches in the codebook and renormalize the augmented dimensions of their corresponding descriptors so that patches belonging to close facial parts can be aggregated into the same codewords.



(a) (b)

Figure 3.4: LFW: averaged variance of Gaussian components on augmented location dimensions vs. (a) window side length with zero offset and vs. (b) standard deviation of window offsets (window side length 200). As the window spatial uncertainty increases, the learned GMM increases the variance of Gaussian distributions on location dimensions, which essentially reduces the influence of location information on codeword assignment.

Since the augmented dimensions are spatially sensitive, they should not be involved in learning the descriptor and codeword relevance distributions. As a result, we use the appearance-based dimensions (first 64 dimensions) of each Gaussian mixture component when computing the relevance weights of codewords and descriptors. Once patches are selected, the last two augmented dimensions of corresponding descriptors are reduced by their mean values, i.e., $[x'_{\text{aug}}, y'_{\text{aug}}] = [x_{\text{aug}} - \bar{x}_{\text{aug}}, y_{\text{aug}} - \bar{y}_{\text{aug}}]$, and the updated descriptors are used in feature aggregation and encoding.

3.9 Experiments

We validate our approach on three face datasets with different foci: (a) image based face verification (b) video based face verification and (c) mobile based face

authentication. In the first two datasets, we perform random shifts to the detected face bounding box to compare the spatial robustness of the original Fisher vector encoding and the proposed selective Fisher vector encoding.

3.9.1 Image based face verification

Labeled faces in the wild (LFW) [73] is an image based face verification dataset. The dataset contains 13,233 images of 5,749 celebrities. The evaluation set is divided into 10 disjoint splits each of which contains 600 image pairs. Of these 300 are positive pairs describing the same person and the other 300 are negatives representing different identities. Two protocols are used for the benchmark: restricted and unrestricted. The restricted protocol prohibits using any outside data for training the models while the unrestricted version allows that. We validate our framework on the restricted protocol to show its performance with limited access to training data.

3.9.1.1 Perturbation generation

To study the sensitivity of localization, we randomly shift the annotated face centers (which are detected by Viola-Jones detector) using a Gaussian distribution $N(0, \sigma^2)$ where σ is chosen from 0, 25, 35 and 50 pixels. We set the window side length to 200 pixels, around 1.7 times the size of the tight facial bounding box.

3.9.1.2 Evaluation

Performance is evaluated using true positive rates at equal error rate (TPR@EER) averaged over the 10 splits. The codebook is trained using perturbed images with 512 Gaussian mixture components. For selective encoding, codeword relevance distributions are learned using 150×150 windows at the face center detected by Viola-Jones detector in the training set. It is worth noting that these windows do not tightly bound the faces.

3.9.1.3 Comparison with original Fisher vectors

Comparison with the original Fisher vectors is shown in Figure 3.5 using both appearance and augmented descriptors. The proposed selective encoding outperforms conventional Fisher vectors using both ℓ^2 metric and diagonal metric learning with 64-D PCA-SIFT descriptors. Interestingly, our method performs better even when there is no centroid perturbation. This might be because even the true facial bounding box includes a small number of distractive patches from the background.

With augmented descriptors, a 1% performance drop of our framework is observed with no center offset using ℓ^2 metric. However, this performance gap vanishes using diagonal metric learning. Our approach also produces more stable performance across multiple levels of window offsets.

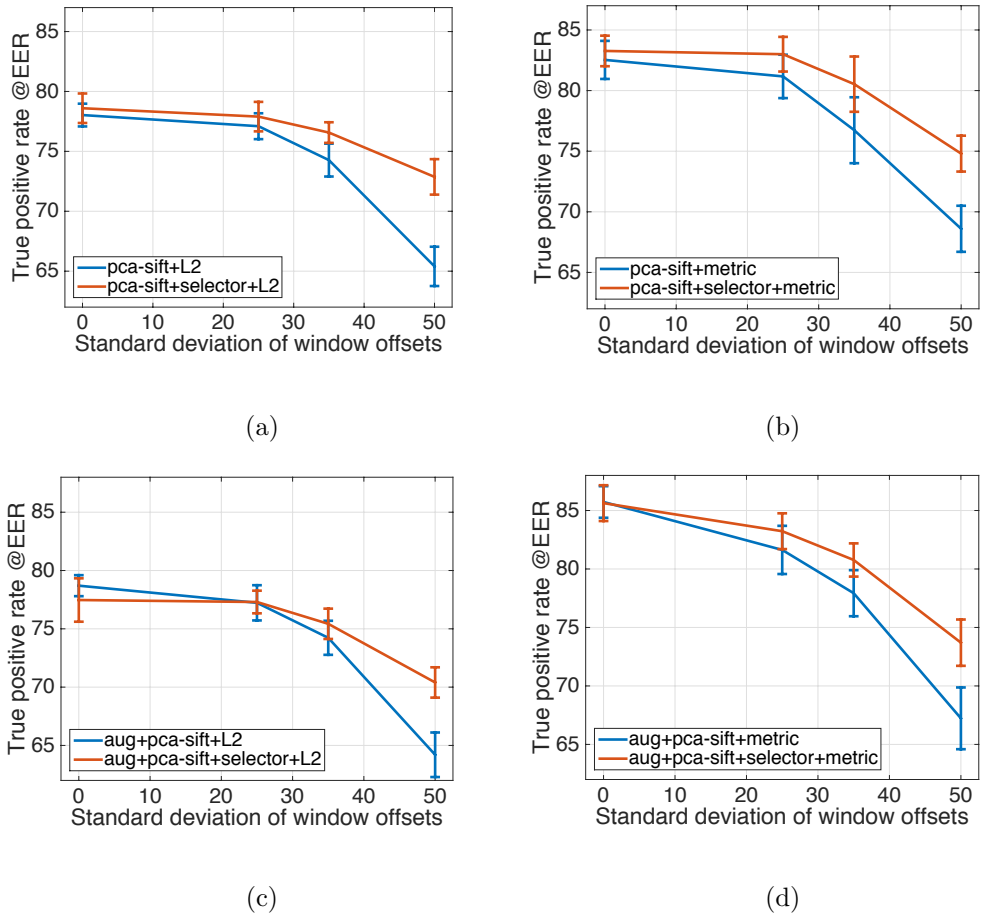


Figure 3.5: LFW: Original FV vs. hard selective FV encoding with PCA-SIFT descriptors with (a) ℓ^2 and (b) diagonal metric learning; original FV vs. soft selective FV encoding with *augmented* descriptors with (c) ℓ^2 and (d) diagonal metric learning.

3.9.1.4 Comparison with perfect face localization

Since our goal is to make the original encoding technique more robust to localization, we compare our framework with the ideal case, where the ground truth face bounding box is known (this will serve as an upper bound on performance, since localization will be perfect). The results with both PCA-SIFT and augmented descriptors are shown in Figure 3.6, where under ℓ^2 metric there is less than 0.5%

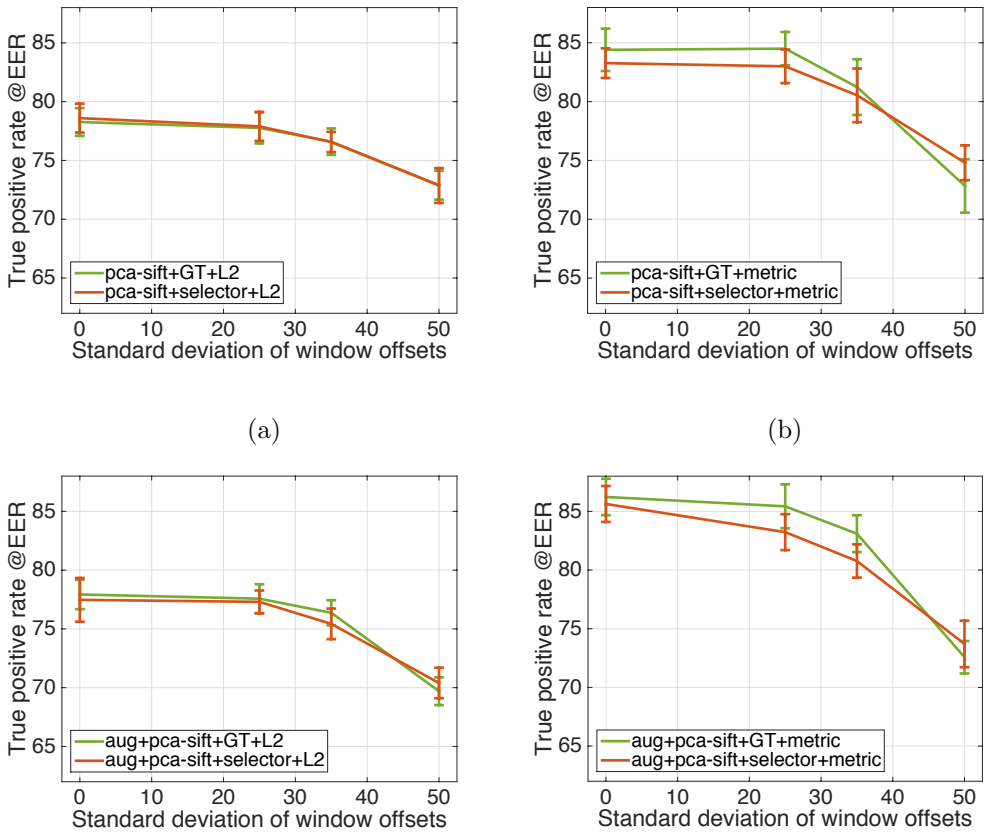


Figure 3.6: LFW: Hard selective FV encoding on perturbed images vs. Original FV encoding on ground truth facial windows with PCA-SIFT descriptors with (a) ℓ^2 and (b) diagonal metric learning; and Soft selective FV encoding on perturbed images vs. FV encoding on ground truth facial windows with *augmented* descriptors with (c) ℓ^2 and (d) diagonal metric learning.

difference between our approach and the ideal one. A larger gap is seen with diagonal metric learning. The ideal case is about 2% better with offset $\sigma = 0, 25, 35$; our approach performs better when more severe face occlusions occur with offset $\sigma = 50$.

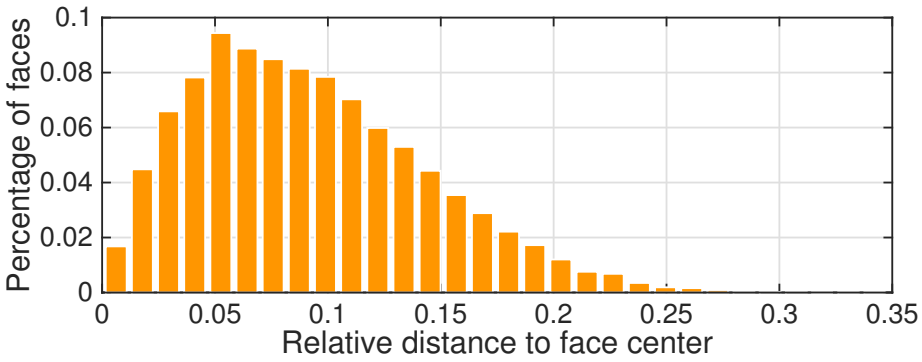


Figure 3.7: LFW: Histogram of relative distances (window side length equals 200 and standard deviation of offset 25) between mean of selected patch locations to true face centers.

3.9.1.5 Appearance-only vs. augmented descriptors

Fisher vectors are usually computed over descriptors augmented with their spatial coordinates, encoding spatial structures into the feature representation. These coordinate features are spatially sensitive and not suitable for learning foreground / background distributions. However, our framework can adapt to such spatially sensitive features by "re-centering" selected patches. Figure 3.7 shows the relative distance between the true face center and the mean coordinates of those selected patches when the window side length is 200 and offset standard deviation is 25. The peak error is around 5% (i.e., 10 pixels).

Our experiments suggest that, compared to appearance-only descriptors, the spatially augmented descriptors perform better with low spatial uncertainty (85.63 ± 1.53 vs. 83.27 ± 1.26 with zero offset and 200 window side length) and gradually degrades with similar performance when the spatial uncertainty increases ($80.77 \pm$

1.42 vs. 80.53 ± 2.28 with 35 offset standard deviation and 200 window side length).

3.9.2 Video based face verification

Youtube Faces (YTF) [74] is a benchmark for video based face verification. The dataset contains 3,425 videos for 1,595 celebrities collected from YouTube movies. All of the faces are localized by the Viola-Jones face detector. The evaluation set is composed of 5,000 pairs of tracks which are also divided into 10 splits. In each split, 250 pairs are positive and the other 250 are negative. For each of the 10 runs, 9 splits are used for training and the remaining split is used for testing. Similar to LFW, the dataset has restricted and unrestricted protocols. Our experiment adopts the restricted protocol in which only 4,500 pairs of videos are available for training the model and the similarity metric.

3.9.2.1 Data preparation

Youtube Faces contains a set of original video frames (faces and background) and a set of cropped and registered face videos. We randomly shift the annotated centers of the faces on each of original videos obeying a uniform distribution $U[-s_{\text{offset}}W, s_{\text{offset}}W]$ in both x and y directions to guarantee that perturbed images have intersections with detector bounding boxes, where s_{offset} is a scale factor and W is the side length of the detected facial bounding box, which differs from person to person. We choose the scale factor s_{offset} among values 0, 0.25, 0.5 and 0.75. For the scale of the windows, we enlarge the side length with another scale factor chosen

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

Figure 3.8: Sample perturbed face images in Youtube Faces dataset: $(\mu_{\text{scale}}, \sigma_{\text{scale}}, s_{\text{offset}})$ = (a) (1, 0, 0), i.e., labeled face bounding box, (b) (2, 0, 0), (c) (2, 0, 0.5) and (d) (2, 0.5, 0.5).

from a Gaussian distribution $N(\mu_{\text{scale}}, \sigma^2_{\text{scale}})$. The mean μ_{scale} is chosen between 1 (original size) and 2 (double size). The σ_{scale} values are chosen from 0, 0.25 and 0.5. We resize all of the perturbed windows to 150×150 for feature encoding. Sampled perturbed images are shown in Figure 3.8.

3.9.2.2 Evaluation

Verification accuracy is also evaluated using TPR@EER, averaged over 10 splits. We downsample each video to 5 frames long. It is worth noting that increasing the sample rate to 20 frames per video produces only 0.04% higher TPR@EER (80.88%) on tightly bounded detected faces than 80.84% obtained from sampling 5 frames per video. Following [68], we apply the incremental "video pooling" for encoding each video, i.e., patch descriptors across frames from the same video are pooled together before being encoded into one Fisher vector. We train PCA and GMM using perturbed training images and learn codeword relevance distributions using detection bounding boxes in sampled training frames for each split.

3.9.2.3 Result

The results comparing the proposed selective encoding and the original Fisher vectors are shown in Tab. 3.1, with different configurations of window scale and offset uncertainty. Both methods use the augmented descriptors and the selector in our approach is trained with soft assignment and tested with no codewords discarded. The results show that our approach outperforms the original Fisher vectors in all settings. Even for the true detected face windows ($\mu_{\text{scale}} = 1, \sigma_{\text{scale}} = s_{\text{offset}} = 0$), our approach obtains slightly improved accuracy. Both approaches experience a 3% performance drop when μ_{scale} is increased from 1 to 2, which is due to the decrease in face resolution, and a 2% drop when σ_{scale} increases from 0.25 to 0.5 with no window offset. Fortunately such high scale uncertainty is typically rare for face detectors and mobile applications. When the scale uncertainty ranges between 0 and 0.25, the encoding quality is relatively stable. The performance gap between the two approaches becomes larger when offset uncertainty increases (over 3% gain when $\mu_{\text{scale}} = 2, \sigma_{\text{scale}} = 0.25, s_{\text{offset}} = 0.75$).

3.9.3 Active face authentication on mobile devices

The use of mobile devices has increased dramatically over the last decades. The privacy protection of mobile phone users has always been an important problem. Verifying the faces recorded by the smartphone camera plays a central role in identifying the users. However, authentication is passively performed in the background, and users may not be actively trying to ensure that their face is viewed

Table 3.1: Youtube Faces: TPR@EER averaged over 10 folds for different perturbation settings using augmented PCA-SIFT descriptors and diagonal metric learning, comparing the proposed selective encoding with original Fisher vectors. Each row represents a setting of face window scaling and relative centroid offset distributions. The better result for each setting is annotated in **bold**.

μ_{scale}	σ_{scale}	s_{offset}	Original FV	Selective FV
1	0	0	$80.84 \pm \mathbf{1.91}$	$\mathbf{81.00} \pm 2.32$
2	0	0	76.72 ± 3.33	$\mathbf{77.24 \pm 2.02}$
2	0	0.5	74.52 ± 1.81	$\mathbf{76.96 \pm 1.73}$
2	0.25	0	76.84 ± 2.27	$\mathbf{77.40 \pm 1.53}$
2	0.25	0.25	$75.04 \pm \mathbf{1.92}$	$\mathbf{77.72} \pm 2.40$
2	0.25	0.5	$74.44 \pm \mathbf{1.26}$	$\mathbf{75.76} \pm 2.08$
2	0.25	0.75	69.64 ± 1.87	$\mathbf{72.88 \pm 1.60}$
2	0.5	0	74.52 ± 1.90	$\mathbf{75.32 \pm 1.60}$
2	0.5	0.5	$70.92 \pm \mathbf{1.35}$	$\mathbf{72.72 \pm 2.07}$

clearly by the camera. This results in face videos with unconstrained poses, some of which are raised faces because users are likely to read while their smartphones are below their faces instead of looking directly at the phone.

3.9.3.1 Dataset

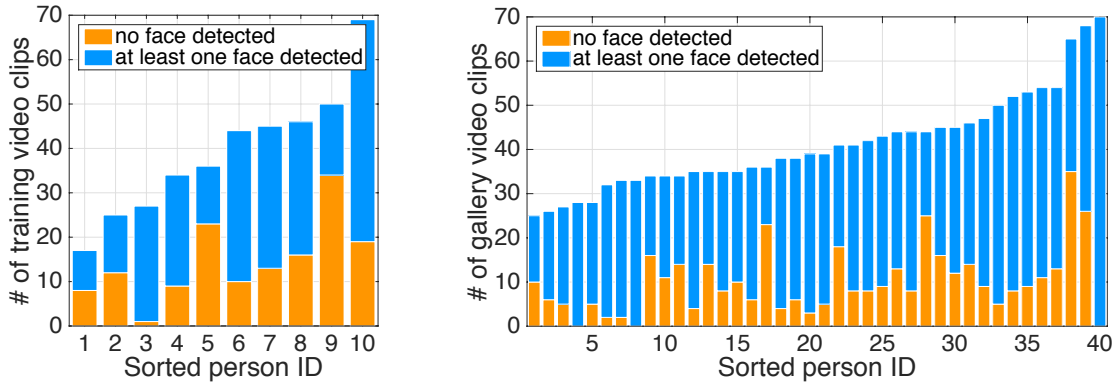
We validate our approach on a dataset that contains 750 long videos recorded from the viewpoint of mobilephone cameras when user activities are present [76]. More specifically, there are 50 persons (subjects) participated in the video recording. Each subject is asked to use the same smartphone to perform 5 different tasks, i.e., Enrollment, Scrolling, Popup, Picture and Document, under three different lighting conditions, i.e., well-lit, dim-lit and natural. The Enrollment task is to ask the user to record their faces in different poses and this data will be the gallery in the face

verification protocol. All the other four tasks involve the users performing some activities on the cellphone (refer [76] for details); these videos make up the probe set.

In practice, it is sufficient to identify users every few seconds. So we sample 30 short clips, each 30 frames long (approximately one second) for each test video. For the gallery set, each enrollment video is segmented into consecutive clips of 30 frames uniformly instead of random sampling. We use the Enrollment data of 10 persons for training and use those of the remaining 40 persons for constructing the gallery set. The lengths of enrollment videos vary for different persons. Figure 3.9 shows the distribution of the training videos and the gallery. Eventually, we have a training set of 393 video clips and a gallery set that contains on average 43 video clips per person. The probe set contains 4 tasks for each person out of 40 for each of the 3 illumination conditions, i.e., 360 video clips per person and 14,400 in total.

3.9.3.2 Evaluation

The evaluation protocol is different from LFW and YTF datasets because, for face authentication, each device has access to only the videos of the owner. So during test time, only the gallery of the corresponding identity is accessible. More specifically, each test clip is compared to all the gallery clips of the corresponding person and a maximum similarity score is calculated. Thereafter, an ROC curve can be generated either by averaging over identities with independent similarity score



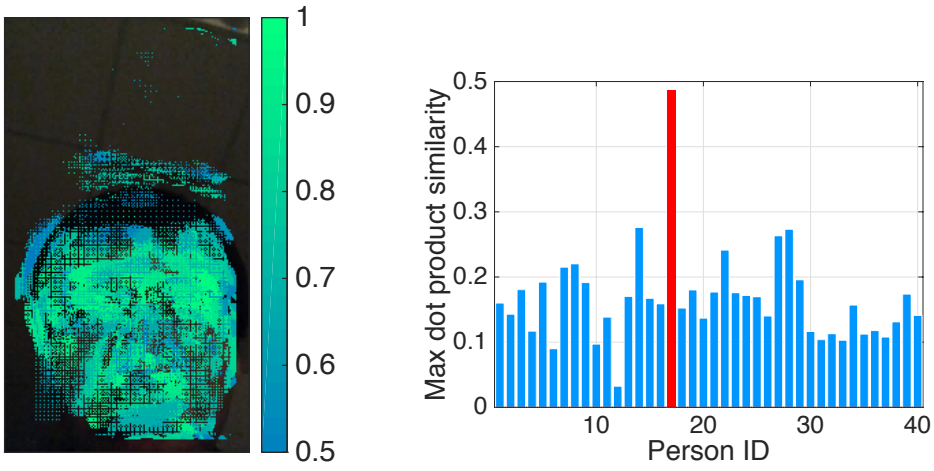
(a) (b)

Figure 3.9: Distribution of the video numbers in (a) training set and (b) gallery set. Identities are sorted in ascending order of their video numbers. Orange bars show the number of videos with no face detected at any of their frames and the blue bars show the number of those with at least one face detected. The training set contains 393 videos in total and the gallery set contains in average 43 videos clips per person.

thresholding or by using a global similarity threshold for all persons. According to our experiments, there is no significant difference between using person-specific thresholds and using a global threshold. So, in all of our experiments, we use global thresholding for ROC curves. Equal error rates (EER) are also used for performance evaluation and comparison.

3.9.3.3 Result

We use the training clips which cover only 10 identities (Figure 3.9) for training PCA and GMM of SIFT descriptors. Also we use all of the images with detected faces in the training set for learning the relevance distribution for selective encoding. Sometimes, real applications may not have large amount of data available for



(a) (b)

Figure 3.10: Probe identity #17: (a) patch centers with relevance (color annotated) larger than 0.5 are shown on top of the origin image and (b) max dot product similarity scores between the Fisher vector of selected patches and that of each gallery video clip. Red color shows the similarity for the ground truth identity.

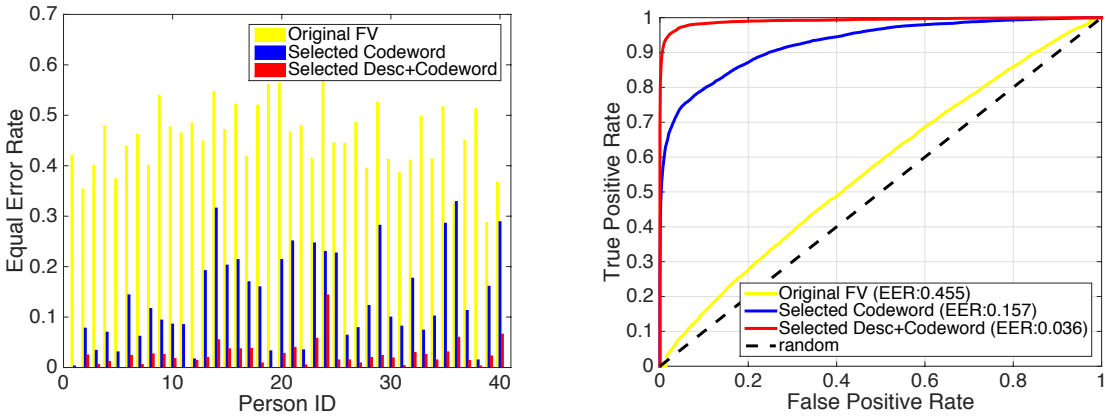
training. So we use such limited training data to evaluate the generalization ability of our trained selector. This experiment is based on appearance descriptors without location features.

We first run an example experiment on a sampled video frame from identity #17. The frame is taken under dark lighting condition and the chin of the identity is slightly out of sight. We apply the selector to dense multi-scale descriptors extracted over the image and obtain for each descriptor a relevance weight. The centroids of patches with higher than 0.5 relevance are plotted on top of the original image in Figure 3.10(a). Most patches within the facial area are selected, although we still see a few background patches selected above the face on the ceiling. These incorrectly selected patches have an insignificant influence on the descriptor distribution when

pooled with a large number of facial patches. We use these selected patch descriptors and the selected codewords (with 0.5 relevance or higher) for encoding the image and compare the feature representation with those from the 40 gallery sets using dot product similarity (equivalent to ℓ^2 since features are normalized). Similarity scores are shown in Figure 3.10(b). The top scored identity is the ground truth and its score is over 0.2 larger than that of the second most similar identity which shows that even using such a dark and low quality image, we are still able to distinguish the identity from all other 39 identities.

The face authentication results are shown in Figure 3.11. We compare our selective encoding framework (based on hard assignment selector) with the original Fisher vectors and a variant of our framework which discards only the codewords with relevance weights lower than 0.5. While the original Fisher vectors achieve 0.455 equal error rate, our approach improves significantly and achieves 0.036 equal error rate. Using only codeword selection achieves 0.157 equal error rate. That means the codeword selection is useful; however the selection of visual descriptors plays a more central role in robustifying feature encoding.

It is worth noting that the detector used for learning the relevance distribution is not specifically tuned in this experiment, so it might still produce errors. However, the experimental results suggest that our selection strategy is robust and does not require accurate registration.



(a) (b)

Figure 3.11: Results on active authentication dataset: (a) Equal error rate (EER) for each person and (b) ROC curves. Three approaches are compared: the original Fisher vector (Original FV), selective encoding with only codeword selection (Selected Codeword), selective encoding with both descriptor selection and codeword selection (Selected Desc+Codeword).

3.10 Discussions and future work

3.10.1 Feature selection

Our work can be seen as a case of doing feature selection. However, typically feature selection is only adopted in the features. Our framework not only selects features but also selects codewords. Another major difference is that the relevance weights of features and codewords are learned from an external weak supervision, i.e., bounding boxes annotated by an existing face detector.

3.10.2 Uncertainty modeling of the codebook

The soft weighted version of our model can also be seen as a special case of modeling the uncertainty of each codeword in the codebook. The weights correspond to how much corresponding codewords contribute to the spatial robustness of the model, i.e., noisy codewords are assigned with much lower weights (high uncertainty).

3.10.3 Deep learning based approaches

There has been a substantial body of work on training deep neural networks for face recognition. However, deep neural nets usually require a large amount of training data, while our approach does not have such requirement and all our experiments are conducted on small training sets, which are insufficient for training neural nets. On the other hand, most existing neural net-based face recognition systems are designed for cropped and aligned face images such as Deepface [20]. Little effort has been devoted to analyzing the spatial robustness of those networks, which could be another interesting story beyond the scope of this paper. Additionally, improving the spatial robustness of neural nets from the perspective of architecture design could be another fruitful topic for future research. A simple approach that utilizes a similar idea of this work in deep neural network would be training a multi-task model that jointly classifies and localizes facial regions. However, the problem of how to train deep nets with limited data and resources is still a big challenge.

3.11 Conclusion

We have proposed a generic selective encoding framework for representing objects of interest that are unreliably localized in images. Our framework introduces the selector component into the codebook model so that it does not require test time detection or registration and becomes robust to localization errors in real scenarios. Our method is also computationally efficient which can benefit real-time applications. We have applied selective encoding to general face verification and mobile phone face authentication. Experimental results suggest that our approach is able to improve the spatial robustness of feature encoding when face detectors produce errors or even fail to localize faces. We expect that our framework could be applied to general image classification and object recognition in the future.

Chapter 4: Generating mid-level spatial representations from language

4.1 Introduction

Text-based image retrieval, dating back to the late 1970s, has evolved from a keyword-based task to a more challenging task based on natural language descriptions (e.g., sentences and paragraphs) [89–91]. Queries in the form of sentences rather than keywords refer to not only object categorical information but also interations, such as spatial relationships, between objects. Those relationships are usually described in the real (3D) world due to the nature of human language. Intuitively, they can be the core feature for ranking images in many application scenarios, e.g., a user searching for images that are relevant to a particular mental image of a room layout. Not surprisingly, researchers have recently increased their focus on understanding spatial relationships from text input and retrieving semantically consistent visual information [89, 92–94].

Matching images with user provided spatial relations is challenging because humans naturally describe scenes in 3D while images are 2D projections of the world. Inferring 3D information from a single image is difficult. Most existing approaches

learn from annotated data to map language directly to a probability distribution of pairwise relationships between object locations [89,92]. However, such a distribution is non-convex and highly non-linear in the 2D image space because the (unknown) camera view affects the bounding box configurations. Consequently, the success of 2D learning based approaches naturally depends on the size of annotated training data. Also, the learner overfits easily since annotated spatial relations have a long-tailed distribution; many valid configurations happen rarely in the real world (e.g., a desk on another desk). With pairwise relations, it is also hard to enforce the fact that all objects are viewed from the same direction in an image. This argues for a holistic model for object relationships that jointly optimizes object configurations. Motivated by this, we explore an alternative model of spatial relations that generates 3D configurations explicitly based on physics.

We explore an approach that uses physical models and complex spatial relation semantics as part of an image retrieval system that generates 3D object layouts from text (rather than from images) and performs image retrieval by matching 2D projections of these layouts against objects detected in each database image. Our framework requires the a priori definition of a fixed set of object and spatial relation categories. Spatial relation terms are extracted from the dependency tree of the text. Objects are modeled using cuboids and spatial relations are modeled as inequality constraints on object locations and orientations. These inequality constraints can become very complex, containing nonlinear transformations represented using first order logic. Consequently, an interval arithmetic based 3D scene solver is introduced to search for feasible 3D spatial layout solutions. Camera orientations are

constrained and sampled for obtaining 2D projections of candidate scenes. Finally, images are scored and ranked by comparing object detection outputs to a sampled set of 2D reference layouts.

Compared to 2D learning based approaches, our approach has the following advantages: (1) the mapping from language to 3D is simple since the text-based spatial constraints have a very concrete and simple meaning in 3D, simple enough to define with a few rules by hand; (2) no training data is needed to learn complex distributions over the spatial arrangement of 2D boxes given linguistic constraints (the non-linear mapping from language to 2D is handled by projective geometry) and (3) adding common sense constraints is easy when referring to physical relationships in 3D (Section 4.6.2), while it is hard if these constraints are specified and learned in 2D (due to the non-linearity of projective geometry). We evaluate our approach using two public scene understanding datasets [95, 96]. The results suggest that our approach outperforms baselines built upon object occurrence histograms and learned 2D relations.

4.2 Related work

Text-based image retrieval has been studied for decades [91]. As both computer vision and natural language processing have advanced, recent efforts have emerged that build connections between linguistic and visual information [97, 98]. Srivastava and Salakhutdinov [99] extend Deep Boltzman Machines (DBMs) to multimodal data for learning joint representations of images and text. They apply such

representations to retrieving images from text descriptions. Their model learns mappings between objects with attributes and their corresponding visual appearances; however spatial relations are not modeled.

Spatial relationships play an important role in visual understanding. Previous works make use of text-extracted spatial relations in image retrieval. Zitnick et al. [94] generate and retrieve abstract cartoon images from text. Cartoon object models are pre-defined and 2D clipart images are composed according to the text. Siddiquie et al. [100] devise a multi-modal framework for retrieving images from sources including images, sketches and text by jointly considering objects, attributes and spatial relationships, and reducing all sources into 2D sketches. However, their framework handles text with only two or three objects and very limited 2D spatial relationships. Lin et al. [92] retrieve videos from textual queries. A set of motion text is defined with visual trajectory properties and parsed into a semantic graph to to match video segments via a generalized bipartite graph matching. All these works rely on 2D spatial relations while our work is based on real world physical models of 3D scenes to retrieve semantically consistent images.

Interesting recent work on retrieving images from text is based on the scene graph representation [89, 93]. A scene graph is a graph-based representation which encodes objects, attributes and object relations. In Johnson et al. [89], text input is converted to a scene graph by a human and a CRF model is used to match scene graphs to images by encoding global spatial relations of objects rather than only pairwise relations. Their approach requires learning spatial relations from annotated image data. Our work differs in that we take a generative perspective and inject

physical relation models and human knowledge into the retrieval system without the requirement of large-scale data annotation.

Many existing works utilize 3D geometry in vision tasks such as object recognition [101], image matching [36], object detection [102,103], *etc.* However, to the best of our knowledge, the use of 3D geometry in relating images with language has not been exploited. While inferring the 3D structure from a single image is challenging and complicated in vision [95, 104–107], the problem of rendering scenes from text is of interest in the graphics community. The wordseye system [108] renders scenes from text with given 3D object models. Chang et al. [109] generates 3D scenes from text by incorporating the spatial knowledge learned from data. In addition, some recent works cast computer vision as inverse graphics and try to incorporate computer graphics elements into visual understanding systems [110–112]. Our work also involves scene generation. However, our purpose is to retrieve similar images based on bounding boxes, which can be efficiently computed using off-the-shelf software during a database indexing step, so real object models are not required, although better scene generation could potentially improve image retrieval accuracy.

4.3 Preliminary – Interval Analysis

Our approach involves finding feasible solutions to a mathematical program where the variables are object coordinates and orientations, and the constraints are inequalities translated from user descriptions. Since small placement pertubations usually do not affect the fullfilment of constraints, feasible variables can naturally

be represented by a set of intervals (any value within the interval is feasible).

Interval analysis represents each variable by its feasible interval, e.g., $[l, u]$ (with lower bound l and upper bound u) and the goal is to find the bound for each dimension that satisfies all constraints [113]. When an interval does not satisfy all the constraints, it is split into smaller intervals and evaluated recursively. Arithmetic operators are defined in terms of intervals, e.g.,

- *addition*: $[l_1, u_1] + [l_2, u_2] = [l_1 + l_2, u_1 + u_2]$;
- *subtraction*: $[l_1, u_1] - [l_2, u_2] = [l_1 - u_2, u_1 - l_2]$;
- *comparison*: $[l_1, u_1] < [l_2, u_2]$ equals $[0, 0]$ if $u_2 \leq l_1$ (definitely false); equals $[1, 1]$ if $u_1 < l_2$ (definitely true); equals $[0, 1]$ otherwise (maybe true).

The fulfillment of a constraint can be represented by any of the three logical intervals, i.e., $[0, 0], [1, 1], [0, 1]$.

4.4 Approach overview

The proposed framework, as illustrated in Figure 4.1, consists of several modules. First, the input text is parsed into a set of semantic triplets of object names and their spatial relationships. Second, the semantic triplets are used to solve possible 3D layouts of objects along with sampled camera locations and orientations. The 2D projections of the 3D scenes are used for generating 2D bounding boxes of objects, which we call *reference configurations*. Finally, the reference configurations are matched to the detected bounding boxes in each database image to score and rank according to their configuration similarity.

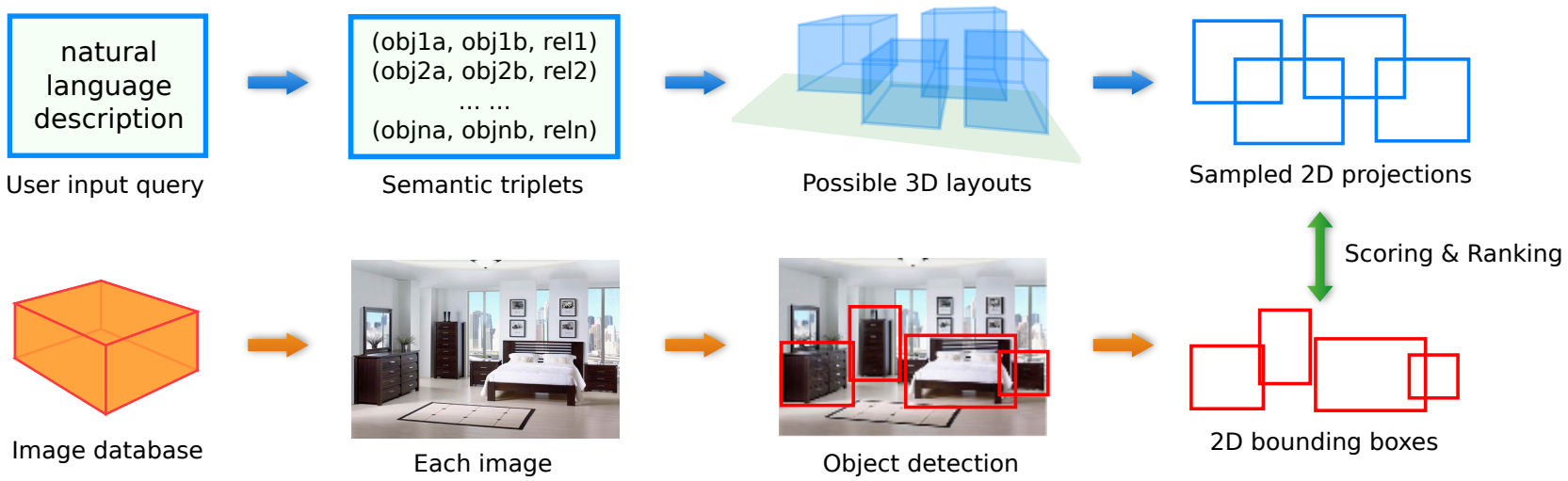


Figure 4.1: Framework overview: a textual description of the visual scene is parsed into semantic triplets which are used for solving feasible 3D layouts and their 2D projections as *reference configurations*. An object detector runs over each database image and generates a 2D bounding box layout, to be matched to reference configurations. All database images are ranked according to their configuration scores.

4.5 Text parsing

The text parsing module translates text into a set of semantic triplets which encode the information about two object instances and their spatial interactions. How to robustly extract relations from text is still an open research problem in natural language processing [90], which is beyond the scope of this paper. For our application, a simple rule-based pattern matching works sufficiently well, requiring a pre-defined dictionary of object and spatial relation categories. A text example and its parsing output is shown in Table 4.1.

The input text is processed by the Stanford CoreNLP library [114] with part-of-speech tagging and dependency tree. We implement a rule-based approach to extract spatial relations (such as *on*, *under*, *in front of*, *behind*, *above*, etc.) from the dependency tree and compose its corresponding semantic triplet representation *(target object, reference object, relation)*. The co-reference module in the CoreNLP library is used to aggregate multiple noun occurrences that correspond to the same object instance. Each object reference is represented by its category name and a unique ID within the category, e.g. `sofa-0` and `dining-table-2`.

Natural objects are usually composed of multiple sub-objects and there are often cases when a sub-object is referenced instead of the whole object. A bed, for instance, has its head and rear. And a chair has its back and seat. We take sub-objects into consideration and represent any sub-object reference by its object category name, unique in-category ID and sub-object name, e.g. "the rear of the bed" is represented as `bed-0:rear` if the ID is 0.

Table 4.1: Semantic triplet parsing from an example query

#	**Sentence → (object-1, object-2, relation)**
1	A picture is above a bed. `(picture-0, bed-0, above)`
2	A night stand is on the right side of the head of the bed. `(night-stand-0, bed-0:head, right)`
3	A lamp is on the night stand. `(lamp-0, night-stand-0, on)`
4	Another picture is above the lamp. `(picture-1, lamp-0, above)`
5	A dresser is on the left side of the head of the bed. `(dresser-0, bed-0:head, left)`

Besides object categories and spatial relationships, we also consider the count of each object, e.g. three chairs, two monitors, etc. The parser maintains a list of object ID and their counts. If the count of `chair-0` is 3, then the parser will expand `chair-0` to a set of three instances {`chair-0-0,chair-0-1,chair-0-2`} in the outputs.

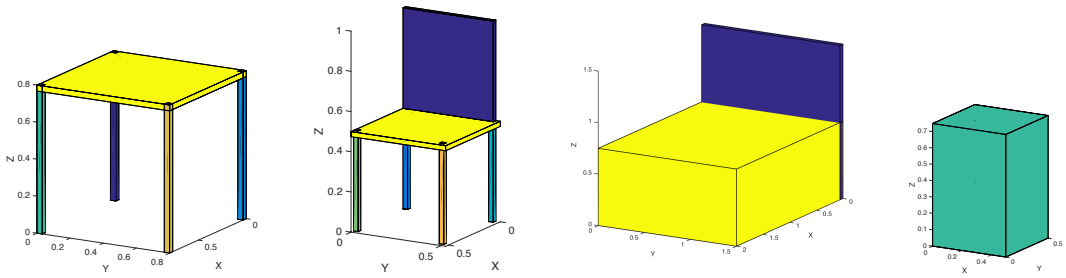
4.6 3D abstract scene generation

The 3D abstract scene generation module is the central component in our image retrieval framework; it takes as input semantic triplets and generates a set of sampled possible 3D object layouts. We describe below the three core components of the scene generator: the cuboid based object model, the spatial relation model and the 3D scene solver.

4.6.1 Cuboid based object model

The basic *cuboid representation* of an object is $\mathbf{C} = (l_x, l_y, l_z, z_s)$ where (l_x, l_y, l_z) is the size of the cuboid that bounds the object in x, y, z directions respectively and z_s is the z-coordinate of the supporting surface of the object. We mostly use regular sizes but also set different sizes for objects with attributes such as `long-desk`, `triple-sofa`, etc. The supporting surface is usually the top face of the object cuboid, but it can sometimes be located elsewhere with respect to the cuboid, e.g., for a chair it is in the middle of the cuboid. Spatial relations such as `on` and `under` are modeled with respect to the surface of the object. Most of the objects can be modeled using this cuboid representation such as `garbage-bin`, `picture`, `night-stand`, etc.

However, the single cuboid representation is not sufficient for some object categories such as `chair` and `desk` since the under-surface area is empty. Considering the fact that most objects can be easily decomposed into smaller sub-objects, we represent these object categories as the union of a set of cuboids, which we call a *cuboid set representation*. Each sub-cuboid corresponds to a sub-object and is considered a simple object, whose top face is the supporting surface. The k-th sub-cuboid is represented by $\mathbf{S}^k = (d_x^k, d_y^k, d_z^k, l_x^k, l_y^k, l_z^k)$ where (d_x^k, d_y^k, d_z^k) is the offset from the lowest point of the sub-cuboid to the lowest point of the original object, and (l_x^k, l_y^k, l_z^k) is the size of the sub-cuboid. The sub-cuboid parameters $\mathbf{S}^k$ are computed as functions of the original object parameters $\mathbf{C}$. Four sampled cuboid based object models are visualized in Figure 4.2.



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

Figure 4.2: Sample cuboid based object representations: (a) table (b) chair (c) bed (d) night-stand. Different colors represent different sub-objects. The night stand (d) is represented by a single cuboid.

4.6.2 Spatial relation model

The spatial location and orientation of each object is represented as $\mathbf{X} = (x, y, z, \theta)$ where (x, y, z) is the lowest point of the object cuboid and d is its orientation. The object rotation is around the z-axis.

Atomic relations. We model 8 basic spatial relations using the following mathematical expressions. Given the object pose and its size, the lowest point $\mathbf{p} = (x_p, y_p, z_p)^\intercal$ and highest point $\mathbf{q} = (x_q, y_q, z_q)^\intercal$ of the object cuboid can be computed by rotating the object models w.r.t. the object orientation such that

$$\begin{aligned} \mathbf{p} &= \mathbf{R}_\theta \left[-\frac{l_x}{2}, -\frac{l_y}{2}, -\frac{l_z}{2}\right]^\intercal + \left[x + \frac{l_x}{2}, y + \frac{l_y}{2}, z + \frac{l_z}{2}\right]^\intercal, \\ \mathbf{q} &= \mathbf{R}_\theta \left[\frac{l_x}{2}, \frac{l_y}{2}, \frac{l_z}{2}\right]^\intercal + \left[x + \frac{l_x}{2}, y + \frac{l_y}{2}, z + \frac{l_z}{2}\right]^\intercal \end{aligned} \tag{4.1}$$

where $\mathbf{R}_\theta$ is the z-axis rotation matrix w.r.t. to orientation θ. So an object can be represented using tuple $(\mathbf{p}, \mathbf{q}, \theta)$. Letting the cuboid of object-1 be $\mathbf{O}_1(\mathbf{p}_1, \mathbf{q}_1, \theta_1)$

with support surface z_{s1} and the cuboid of object-2 be $\mathbf{O}_2(\mathbf{p}_2, \mathbf{q}_2, \theta_2)$ with support surface z_{s2}, we define 8 atomic relations as

- `near`: $\mathbf{O}_1 \cap (\mathbf{p}_2 - d_{\text{near}}\mathbf{e}_{\theta_2}, \mathbf{q}_2 + d_{\text{near}}\mathbf{e}_{\theta_2}, \theta_2) \neq \emptyset$;
- `on`: $z_{p1} = z_{s2} \wedge \frac{\mathbf{p}_1+\mathbf{q}_1}{2} \in_{xy} \mathbf{O}_2$;
- `above`: $z_{q2} + d_{\text{min-above}} \leq z_{p1} \leq z_{q2} + d_{\text{max-above}} \wedge \frac{\mathbf{p}_1+\mathbf{q}_1}{2} \in_{xy} \mathbf{O}_2$;
- `under`: $z_{s1} < z_{s2} \wedge \mathbf{O}_1 \cap_{xy} \mathbf{O}_2 \neq \emptyset$;
- `behind`: $\max(\mathbf{u}_{\theta_2}^\intercal \mathbf{p}_1, \mathbf{u}_{\theta_2}^\intercal \mathbf{q}_1) \leq \min(\mathbf{u}_{\theta_2}^\intercal \mathbf{p}_2, \mathbf{u}_{\theta_2}^\intercal \mathbf{q}_2)$;
- `front`: $\min(\mathbf{u}_{\theta_2}^\intercal \mathbf{p}_1, \mathbf{u}_{\theta_2}^\intercal \mathbf{q}_1) \geq \max(\mathbf{u}_{\theta_2}^\intercal \mathbf{p}_2, \mathbf{u}_{\theta_2}^\intercal \mathbf{q}_2)$;
- `on-left`: $\min(\mathbf{u}_{\theta_2-\pi/2}^\intercal \mathbf{p}_1, \mathbf{u}_{\theta_2-\pi/2}^\intercal \mathbf{q}_1) \geq \max(\mathbf{u}_{\theta_2-\pi/2}^\intercal \mathbf{p}_2, \mathbf{u}_{\theta_2-\pi/2}^\intercal \mathbf{q}_2)$;
- `on-right`: $\max(\mathbf{u}_{\theta_2-\pi/2}^\intercal \mathbf{p}_1, \mathbf{u}_{\theta_2-\pi/2}^\intercal \mathbf{q}_1) \leq \min(\mathbf{u}_{\theta_2-\pi/2}^\intercal \mathbf{p}_2, \mathbf{u}_{\theta_2-\pi/2}^\intercal \mathbf{q}_2)$;

where $d_{\text{near}}, d_{\text{min-above}}, d_{\text{max-above}}$ are distance thresholds, $\mathbf{p} \in_{xy} \mathbf{C}$ means point $\mathbf{p}$ is inside the cuboid $\mathbf{C}$ on the x-y plane, $\cap$ represents the intersection of two cuboids and $\cap_{xy}$ the intersection of two cuboids on the x-y plane, and $\mathbf{u}_\theta = (\cos\theta, \sin\theta, 0)^\intercal$ is a unit direction vector and $\mathbf{e}_\theta = (\cos\theta - \sin\theta, \sin\theta + \cos\theta, 1)^\intercal$ is a vector that enlarges the effective object cuboid.

Composite relations. In natural language, there are far more spatial relation descriptions than the above mentioned 8 relations. However, most of the spatial relations can be defined based on the 8 atomic relations. Two examples are

- `next-to`: `on-left`$(\mathbf{O}_1, \mathbf{O}_2) \vee$ `on-right`$(\mathbf{O}_1, \mathbf{O}_2)$;
- `side-by-side`: $\theta_1 = \theta_2 \wedge$ `near`$(\mathbf{O}_1, \mathbf{O}_2)$;

In addition, another relation is modeled which is usually used for a set of multiple instances $\{\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_k\}$ of the same object category, i.e.,

- `in-a-row`: $\theta_i = \theta_{i+1} \wedge \texttt{on-right}(\mathbf{O}_i, \mathbf{O}_{i+1}),\ \forall i.$

Group relations. If an object reference has a count more than 1, then all of its instances form a group, which often interacts with other objects as an entirety. If a group of k instances occurs in the triplet as the target, we create k new triplets with the same reference and relation. If the group occurs as the reference, then we create a new virtual object whose cuboid is bounded by all of its instances.

Prior constraints. An effective way to reduce the search space is to incorporate common sense and reasonable assumptions into the constraints. First, we make the following assumptions: (a) the room has two walls ($x = 0$ and $y = 0$); (b) the text description is coherent, i.e., the objects in each semantic triplet are close to each other; (c) objects are usually oriented along x-axis or y-axis directions. Second, no pair of objects overlap with each other, i.e.,

- `exclusive`: $\mathbf{S}_i^v \cap \mathbf{S}_j^w = \emptyset \forall i, j, v, w$

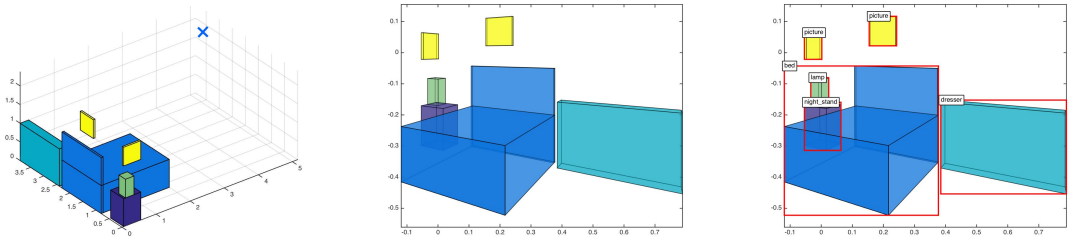
where $\mathbf{S}_i^v$ is the v-th component (sub-cuboid) of the i-th object. Many other constraints are related with object properties: (a) picture, door, mirror are on the wall, i.e. $x = 0 \vee y = 0$; (b) for relation next-to, in-a-row, side-by-side, if either reference or target is against the wall, the other ones are also against the wall and they should also have the same orientation; (c) bed, night-stand, sink are against the wall; (d) bed, night-stand, sofa are on the ground.

4.6.3 3D scene solver

Let $\mathbf{X} = \{x_1, y_1, z_1, \theta_1, \ldots, x_n, y_n, z_n, \theta_n\} \in \mathbb{R}^{4n}$ be a *layout state* representing the locations and orientations of all objects. We construct *constraint function* $F : \mathbb{R}^{4n} \rightarrow \{0, 1\}$ which evaluates all prior constraints and relational constraints. The goal is to find the feasible solution set $\mathbf{S}$ such that $F(\mathbf{X}) = 1$ for all $\mathbf{X} \in \mathbf{S}$.

Our solver is based on interval analysis [113] where any variable is represented by an interval (an uncertain value) instead of a certain value. We use a vector of size 2 to represent an interval, i.e., a lower bound and an upper bound. Under interval analysis, the domain of layout states becomes $\mathbb{R}^{4n\times 2}$ and the constraint function becomes $F : \mathbb{R}^{4n\times 2} \rightarrow \{[0, 0], [0, 1], [1, 1]\}$. Starting with a candidate queue containing an initial interval layout state $\{\mathbf{X}_0\}$, our solver examines the candidate states one at a time. For each state $\mathbf{X}_i \in \mathbb{R}^{4n\times 2}$, if $F(\mathbf{X}_i) = [1, 1]$, then $\mathbf{X}_i$ is feasible and appended to the solution set. If the constraint fullfillment is undecidable, i.e., $F(\mathbf{X}_i) = [0, 1]$, then $\mathbf{X}_i$ is divided into two equally sized intervals by splitting the variable with the largest uncertainty. The two new states are appended to the candidate queue. Otherwise, $F(\mathbf{X}_i) = [0, 0]$ and no feasible solution is within the space bounded by $\mathbf{X}_i$. In the end, any layout in the solution set is guaranteed to meet all constraints. An advantage of the method is that it does not require computing the gradient of constraint F. The pseudo-code is shown in Algorithm 1.

Interval shrinkage. The original interval analysis does not make full use of equality constraints, e.g., when a variable is constrained to equal another variable, it becomes



(a) (b) (c)

Figure 4.3: The generated scene geometry for the query in Table 4.1: (a) a sampled 3D layout with the sampled camera location (a blue cross in the figure), (b) 2D projections of the object cuboids and (c) 2D bounding boxes of the objects.

redundant to divide both of their intervals since one can be directly computed based on the other. In addition, many spatial relations are transitive, e.g., if object A is in front of object B and B is in front of C, then A is likely to be in front of C but with a larger distance. Such inferred constraint can benefit the solver with a better pruning power. Based on these observations, we develop the interval shrinkage operation which pre-computes lower bound matrices $\mathbf{L}^x, \mathbf{L}^y, \mathbf{L}^z \in \mathbb{R}^{n \times n}$ and upper bound matrices $\mathbf{U}^x, \mathbf{U}^y, \mathbf{U}^z \in \mathbb{R}^{n \times n}$ for pairwise coordinate differences, i.e., $L^x_{i,j} \leq x_i - x_j \leq U^x_{i,j} \wedge L^y_{i,j} \leq y_i - y_j \leq U^y_{i,j} \wedge L^z_{i,j} \leq z_i - z_j \leq U^z_{i,j}$. The bound matrices are initialized using the original constraints and updated once we find $L^*_{i,j} < L^*_{i,k} + L^*_{k,j}$ or $U^*_{i,j} < U^*_{i,k} + U^*_{k,j}$ $(* \in \{x, y, z\})$. Before evaluating each candidate interval layout state, we shrink its variables according to the bound matrices, e.g., $\mathbf{x}_i^{\text{shrink}} = \cap_j [x_j + L^x_{i,j}, x_j + U^x_{i,j}] \cap \mathbf{x}_i$ where $\mathbf{x}_i$ is the interval of variable x_i and $\mathbf{x}_i^{\text{shrink}}$ is the interval after shrinkage.

Algorithm 1: 3D scene solver

Data: Initial bounds $\mathbf{X}_0 = [\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_1, \theta_1, \ldots, \mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n, \theta_n] \in \mathbb{R}^{4n\times 2}$
Data: Constraint $F : \mathbb{R}^{4n\times 2} \to \{[0,0],[0,1],[1,1]\}$
Result: Feasible regions (or solution set) $\mathbf{S}$

initialization: $\mathbf{S} = \emptyset, \mathbf{Q} = \{\mathbf{X}_0\}$;
while $\mathbf{Q} \neq \emptyset$ **do**
    read the first interval: $\mathbf{X}_i = \mathbf{Q}$.front();
    remove the first interval: $\mathbf{Q}$.pop();
    interval shrinkage: $\mathbf{X}_i = \text{shrinkage}(\mathbf{X}_i)$;
    if $F(\mathbf{X}_i)$=$[0,0]$ **then**
        $\mathbf{X}_i$ is not feasible;
    else if $F(\mathbf{X}_i)$=$[1,1]$ **then**
        $\mathbf{X}_i$ is feasible: $\mathbf{S}$.append($\mathbf{X}_i$);
    else if $\max_k |X_{ik}.\max - X_{ik}.\min| > tol$ **then**
        $k = \arg\max_k |X_{ik}.\max - X_{ik}.\min|$;
        half split k-th dimension of $\mathbf{X}_i$ into $\mathbf{X}_i^{(1)}$ and $\mathbf{X}_i^{(2)}$;
        $\mathbf{Q}$.append($\mathbf{X}_i^{(1)}$);
        $\mathbf{Q}$.append($\mathbf{X}_i^{(2)}$);
end
return $\mathbf{S}$;

Early stopping. The feasible solution space can be large if the input constraints are weak. Since we sample K layouts in our framework for subsequent image matching, the 3D scene solver stops when at least K layouts are found. The sampling behaviour is achieved by implementing the candidate queue with Knuth shuffling, i.e., each time after appending a new element, the queue randomly pick an element and swaps it with the new element.

The problem is a combinatorial optimization which is NP-hard and interval analysis is essentially a breadth first search with pruning. As a result, the algorithm has no time limit guarantee. However, with interval shrinkage and early stopping, our algorithm is able to solve most queries in a reasonable amount of time. Without interval shrinkage, our MATLAB implementation can not find a solution for the query

in Table 4.1 within 10 minutes, while it returns 5 solutions with only 6 seconds using the shrinkage operation.

4.7 Image retrieval

To compare a query with image bounding boxes, we first sample feasible 3D layouts and potential camera locations and orientations to produce reasonable 2D projections of objects and then compute their bounding boxes. The whole image database is scored and ranked according to the similarity between bounding boxes detected by object detectors and those from sampled 2D layouts.

4.7.1 3D layout sampling

The 3D solver finds (continuous) interval solutions for 3D object coordinates; any solution within such intervals is feasible. However, the solutions within an interval are redundant; those object locations shift in tiny distances. So we sample only one layout within each interval, which results in a set of representative feasible 3D layouts. We further sample a few 3D layouts from this feasible set in order to generate their 2D projections.

4.7.2 2D layout projections

For each layout, we sample camera locations and orientations to obtain 2D projections which allows matching images under multiple views. Object bounding boxes are computed according to the 2D projections. Since we solve for scale and

translation for each image individually during matching, in this step we only consider a canonical camera. Some heuristics are used for sampling camera locations and orientations. First, the camera always faces the objects and should be neither too close nor too far, so we sample its location from 5-10 meters from the origin. Second, the camera should not be located behind the wall, so the coordinates are positive. Third, when an object is on the wall, the camera direction should be within 60 degree offset from the object orientation. We assume the camera is 1.7 meters above the ground and situated horizontally. Figure 4.3 shows an example of 3D layout, 2D projections and 2D bounding boxes for the query in Table 4.1.

4.7.3 2D layout similarity

Both detection outputs and 2D reference layouts can be represented by $\{\mathbf{b}_i, c_i\}$ where $\mathbf{b}_i$ is the 2D box of the i-th object and c_i is its category. Let $\{\mathbf{b}_i, c_i\}$ be a 2D reference layout and $\{\mathbf{b}'_i, c'_i\}$ be the detected boxes. Since scaling and translation are left as free variables, the bounding box matching involves optimizing

$$\max_{s,t,\mathbf{a}} \sum_i p(\mathbf{b}'_{a_i}) \cdot \text{IOU}(s\mathbf{b}_i + t, \mathbf{b}'_{a_i}), \quad s.t.\ c_i = c'_{a_i}, \tag{4.2}$$

where $p(\mathbf{b}'_k)$ is the detection confidence, IOU is intersection-over-union and assignment vector $\mathbf{a}$ indicates the correspondence between two sets of bounding boxes. In our experiment, we evaluate two versions: (a) the *hard* version uses a threshold on detection outputs and uniform $p(\mathbf{b}'_k)$ and (b) the *soft* version makes $p(\mathbf{b}'_k)$ equal to the detection score. We use a sliding window to find the best matched transformation and assignment. Specifically, we uniformly sample 5 scale factors from 0.5 to 1

w.r.t.the image space and search with a 10-pixel stride. We use a greedy strategy to compute assignments and scores (Equation 4.2). The score for a query is computed as the highest score among the scores of all its sampled 2D layouts.

4.8 Experiments

We validate our approach using two indoor scene datasets (SUN RGB-D [96] and 3DGP [95]). Although the original goal of the two datasets is not text-based image retrieval, both contain groundtruth object bounding boxes which enables evaluation in our image retrieval setting. We compare 3 baselines built upon object occurrence histogram and 2D spatial relation based scene graph matching.

4.8.1 Experimental setup

Baseline (H). The first baseline is based on the histogram of object occurrences. Specifically, both the image and text are converted to a histogram representation, i.e., a vector $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$, where x_i is the number of occurrences of the i-th object category. The similarity between occurrence histograms is measured by ℓ^1 distance.

Baseline (2D). The second baseline is based on learned object relations in 2D image space. Specifically, the baseline learns a bounding box distribution of the first object w.r.t.the second object box (normalized in both x and y coordinates). We have all eight atomic relations annotated in 1,000 images in the training set of SUN RGB-D dataset and use IOU-based nearest neighbor (IOU-NN) classifier to score for each

test image the spatial relationships between object pairs. Following [115], we convert the text to a simplified scene graph that maps all instances of an object category into a single node, and assign the count of each relation as an attribute of the corresponding edge. An image scene graph with relation probabilities on edges can be constructed for each test image by using the IOU-NN relation classifier upon each pair of detected object instances. To measure the similarity between text scene graph and image scene graph, we sum for each edge (u, v, r) in the text scene graph the top $k_{u,v,r}$ corresponding relation scores in the image scene graph, where $k_{u,v,r}$ is the count of the relation r between object categories u and v in text scene graph.

Baseline (CNN). The third baseline replaces the IOU-NN relation classifier in Baseline 2D with a Convolutional Neural Network (CNN). Following [116], we finetune the pretrained VGG-19 [31] to predict predicates from cropped union image regions of the two objects. The word2vec vectors of the two objects are concatenated with the response of layer *fc7*. We backpropagate through the whole network with initial learning rate 0.001 for 90 epochs.

Evaluation metric. We evaluate different approaches to retrieving indoor images from text descriptions by measuring the percentage of queries (recall) at least one of whose ground truth images are retrieved within top k ranked images ($R@k$). The median rank (median of the ranks of all ground truths) is used as a global measurement.

Parameter selection. We set the room size to be $5m \times 5m \times 5m$. $d_{\text{near}} = 0.5m$, $d_{\text{min-above}} = 0.25m$, and $d_{\text{max-above}} = 0.5m$. The tolerance in 3D scene solver is $0.2m$ because $20cm$ replacement of objects is unlikely to change the constraint fulfillment. We sample 5 reference layouts per query and 1 camera view per layout unless otherwise specified.

4.8.2 SUN RGB-D dataset with R-CNN detectors

SUN RGB-D Dataset [96] is a recent dataset for scene understanding which contains $10,335$ RGBD images. We use only the RGB images without depth information. We follow the same protocol as [96] by using $5,285$ images for training the detectors and the remaining $5,050$ images as the evaluation set. We annotated text queries for 150 sampled test images. SUN RGB-D contains various objects and complex spatial relations. We choose 19 object categories in our evaluation: {*bed, chair, cabinet, sofa, table, door, picture, desk, dresser, pillow, mirror, tv, box, whiteboard, night_stand, sink, lamp, garbage_bin, monitor*}, which contains not only objects on the floor but also those off the ground or on the wall such as *picture* and *mirror*.

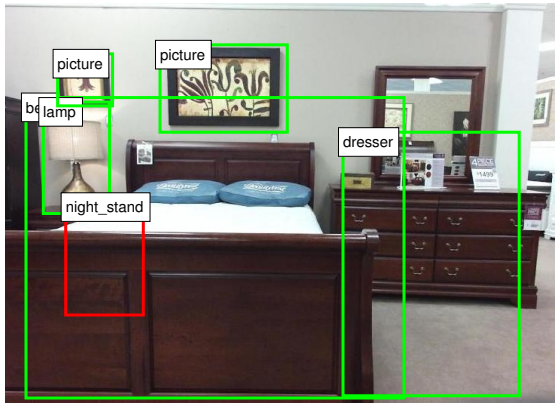
We use the $5,285$ training images and their ground truth object bounding boxes to train Fast R-CNN [117] detectors for the 19 object categories. The R-CNN approach is built upon object proposals; non-maximum suppression is not used in postprocessing. For each test image, R-CNN detectors generate probability-like scores for all object categories on each object proposal bounding box. The category

Table 4.2: SUN RGB-D: Top-k retrieval accuracy for 150 queries. The retrieval candidate set contains 5,050 images. We evaluate the occurrence baseline (H), 2D relation baseline (2D), CNN baseline, the proposed hard version, proposed soft versions, and a combination between our soft version and the 2D baseline. The parameter of our model $[x, y]$ means sampling x 3D layouts and y camera views for each layout. All results of our model are averaged over 5 random trials. The threshold for detection outputs is 0.5. The best is shown in **bold** and the second best is shown with underline.

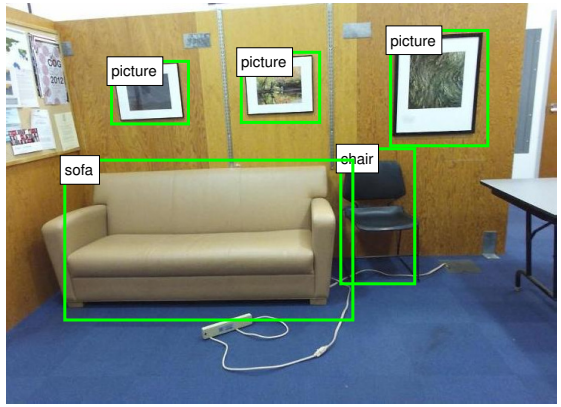
	R@1	**R@10**	**R@50**	**R@100**	**R@500**
Baseline H	1.3	4.0	14.0	20.0	43.3
Baseline 2D	2.7	15.3	35.3	44.0	64.0
Baseline CNN	2.7	16.7	30.7	36.0	63.3
Ours hard[5,1]	3.9	16.4	31.7	42.3	71.7
Ours soft[5,1]	4.5	16.7	34.0	46.4	76.0
Ours soft[5,5]	4.9	18.7	37.9	48.1	76.9
Ours soft[5,5] + 2D	**8.7**	**21.6**	**40.5**	**50.7**	**77.6**

with the highest score is chosen as the bounding box category and its score is used as the bounding box confidence.

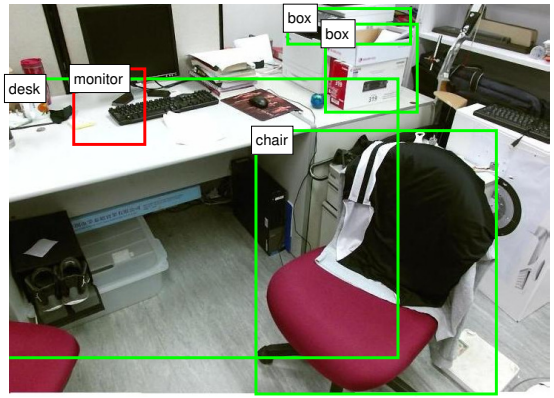
The top-k retrieval recalls are shown in Table 4.2. In addition with the baselines, two versions of our approach are evaluated. The baselines and our hard model use bounding boxes with over 0.5 confidence and weigh them equally, while our soft models use all bounding boxes and assign their confidences as weights in Equation 4.2. The results suggest that the hard model with 5 layout samples outperforms the occurrence baseline and is on par with the 2D baseline. Our soft models perform even better than the hard one. With increased layout samples, our approach outperforms the baselines significantly. We also evaluate a combination between our soft model and the 2D baseline by adding their normalized scores. The result suggests that such combination further boost the accuracy and that our physical model based solution is complementary to learning based approaches.



(a) A picture is above a bed. A night stand is on the right side of the head of the bed. A lamp is on the night stand. Another picture is above the lamp. A dresser is on the left side of the head of the bed.

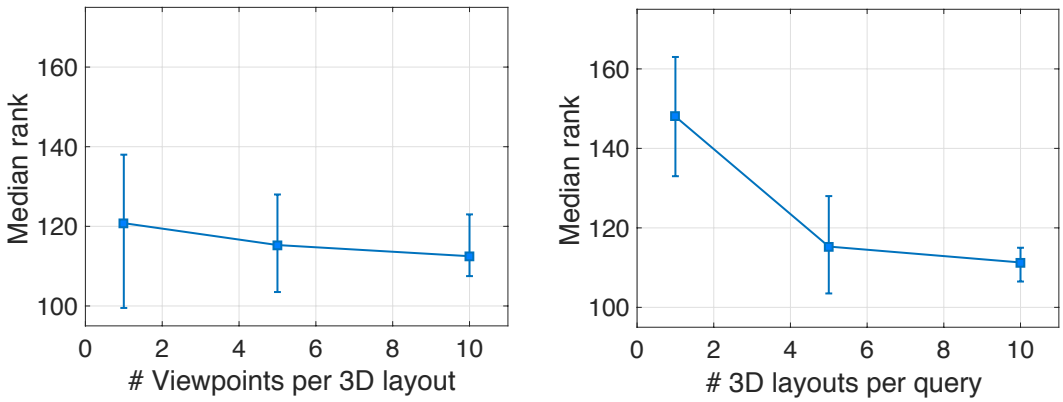


(b) There is a triple sofa. The sofa is against the wall. A chair is next to the sofa. And the chair is also against the wall. Two pictures are above the sofa. And another picture is above the chair.



(c) A chair is in front of the desk. Some boxes are on the desk. A monitor is on the desk. The desk is against the wall.

Figure 4.4: Matched object layouts based on our greedy 2D layout matching for three ground truth images that are ranked top 5 among all candidate images. Green bounding boxes are object detection outputs that match the 2D layouts generated from the text queries. Red bounding boxes represent a missing object (not detected by the object detector) within the expected region proposed by 2D layouts.



(a) (b)

Figure 4.5: Influence of # viewpoint samples and # layout samples: (a) 5 3D layouts sampled for each query, and (b) 5 viewpoint sampled for each 3D layout. The y-axis is median rank of ground truths. We random 5 times for each data point. Lower is better.

Qualitative results. Figure 4.4 shows 3 examples whose ground truths are ranked top 5. The object bounding boxes that best match the generated 2D layouts are shown on the images. Green boxes are matched objects and red boxes are missing ones, expected in the generated 2D layout but unseen in the object detection output. The figure shows that our model has some level of tolerance on missing detections. A more interesting finding is that our model suggests potential locations for missing objects even though they could be heavily occluded.

Sampling effects. To obtain 2D layouts, we sample 3D layouts and camera views. Figure 4.5 shows how the sample size of both affects the the median rank of ground truths (keeping one and varying the other). Figure 4.5 suggests that more samples generally yield better performance and the improvement saturates as the sample

size increases. The improvement brought by more 3D layouts is more significant than that brought by more camera views. In addition, the performance uncertainty due to randomness decreases as the sample size increases.

4.8.3 3DGP dataset with DPM detectors

The 3DGP dataset [95] contains 1,045 images with three scene types: living room, bedroom and dining room. Each image is annotated with bounding boxes for 6 object categories: *sofa*, *table*, *chair*, *bed*, *side table* and *dining table*. Following the same protocol as in [95], 622 training images are used to train the furniture detectors and the remaining 423 images are used as the retrieval image database. We use pre-trained Deformable Part Models (DPM) [26] of indoor furnitures provided by the 3DGP dataset and use the thresholds in the pre-trained models to cut off false alarms. Non-maximum suppression is used to remove duplicates.

3DGP dataset is less diverse than SUN RGB-D; many images have very similar layouts. We annotated 50 unique layout descriptions which cover 222 test images. The retrieval results are shown in Table 4.3. Because our method is agnostic about object detector algorithms, we split the results into two parts to separate the impact from using a specific detection algorithm: one using ground truth bounding boxes and the other using DPM detection outputs. The results suggest that our approach outperforms baseline algorithms under both bounding box settings and the improvement is independent from detector performances.

Table 4.3: 3DGP dataset: Top-k image retrieval accuracy. Left half is based on DPM (the best is with **bold**) and right half is based on ground truth bounding boxes (the best is in underline). The results of our approach (soft[5,5]) are averaged over 10 random trials.

	w/ DPM bbox				**w/ GT bbox**			
	H	2D	CNN	Ours	H	2D	CNN	Ours
R@1	4.0	2.0	4.0	**4.4**	4.0	4.0	4.0	3.0
R@10	10.0	14.0	16.0	**16.8**	16.0	18.0	14.0	20.2
R@50	30.0	30.0	30.0	**31.2**	34.0	38.0	32.0	41.4
R@100	46.0	32.0	32.0	**52.0**	64.0	66.0	66.0	68.0

4.9 Discussions and future work

4.9.1 Generative vs. discriminative

Our work is a generative framework which generates an abstract scene layout from a given textual description. The proposed generative model is based on human common sense about spatial relationships and indoor scene structures. We show our generative approach is effective when there is a lack of training pairs of image and descriptions. We argue that it would be difficult to collect sufficient training pairs in this task considering the diversity of human language descriptions. However, it would still be interesting to see how we could effectively learn a generative model following the recent emerging directions such as Generative Adversarial Network (GAN) models and its variant Conditional GAN models.

Another direction to address this problem would be the discriminative way, i.e., interpreting the spatial relationships or textual phrases directly from images. There has been a recent effort in this problem. However, a major challenge of such

problem is still how to learn the predictive model under a limited size of annotated training data. In the next chapter, we introduce an approach to predicting phrases from images using large scale weakly supervised data.

4.9.2 Diverse solutions and joint optimization

A bottleneck of our framework is the need to sample layout solutions. In terms of sampling, an important problem is how to sample diverse solutions and avoid near duplicates in order to improve the efficiency of sampling process. The interval analysis is essentially the breadth first search. A naive way would be deduplicate the solutions after sampling a large number of solutions. Efficient sampling strategies could benefit the system very much. On the other way around, instead of sampling, an interesting further direction would be exploring how to unify the the whole pipeline by combining top-down and bottom-up approaches so that the need of sampling could be largely reduced. Essentially, this would lead to a joint optimization of the object layouts given both textual descriptions and images.

4.9.3 Nonrigid objects and natural scenes

Our framework works mostly for rigid objects and indoor scenes because the generative model assumes a cuboid based object representation. It would be interesting to see whether and how this idea could be generalized to natural scnes with nonrigid objects and complex relationships. A possible direction to pursue would be representing the objects as probabilistic distributions instead of using concrete

shapes. That would benefit the system by not only allowing more flexibility of the object rigidness but also allowing the interactions between objects more tolerable to small noises or uncertainty.

4.10 Conclusion

We presented a general framework for retrieving images from a natural language description of the spatial layout of an indoor scene. The core component of our framework is an algorithm that generates possible 3D object layouts from text-described spatial relations and matching these layout proposals to the 2D image database. We validated our approach via the image retrieval task on two public indoor scene datasets and the result shows the possibility of generating 3D layout proposals for rigid objects and the effectiveness of our approach to matching them with images.

Chapter 5: Large scale wealy supervised learning of high-level representations

5.1 Motivation

Research on visual recognition models has traditionally focused on supervised learning models that consider only a small set of discrete classes, and that learn their parameters from datasets in which (1) all images are manually annotated for each of these classes and (2) a substantial number of annotated images is available to define each of the classes. This tradition dates back to early image-recognition benchmarks such as CalTech-101 [118] but is still common in modern benchmarks such as ImageNet [119] and COCO [120]. The assumptions that are implicit in such benchmarks are at odds with many real-world applications of image-recognition systems, which often need to be deployed in an *open-world* setting [121]. In the open-world setting, the number of classes to recognize is potentially very large and class types are wildly varying: they include generic objects such as "dog" or "car", landmarks such as "Golden Gate Bridge" or "Times Square", scenes such as "city park" or "street market", and actions such as "speed walking" or "public speaking". The traditional approach of manually annotating images for training does not scale

Figure 5.1: Four high-scoring visual n-grams for three images in our test set according to our visual n-gram model, which was trained *solely* on *unsupervised* web data. We selected the n-grams that are displayed in the figure from the five highest scoring n-grams according to our model, in such a way as to minimize word overlap between the n-grams.

well to the open-world setting because of the amount of effort required to gather and annotate images for all relevant classes. To circumvent this problem, several recent studies have tried to use image data from photo-sharing websites such as Flickr to train their models [122–128]: such images have no manually curated annotations,

but they do have metadata such as tags, captions, comments, and geo-locations that provide weak information about the image content, and are readily available in nearly infinite numbers.

We propose to follow [125] and study the training of models on images and their associated user comments present in the YFCC100M dataset [129]. In particular, we aim to take a step in bridging the semantic gap between vision and language by *predicting phrases that are relevant to the contents of an image*. We develop *visual n-gram* models that, given an image $\mathbf{I}$, assign a likelihood $p(w|\mathbf{I})$ to each possible phrase (n-gram) w. Our models are convolutional networks with output layers that are motivated by n-gram smoothers commonly used in language modeling [130, 131]: for frequent n-grams, the image-conditional probability is very precisely pinned down by trainable parameters in the model, whereas for infrequent n-grams, the image-conditional probability is dominated by the probability of smaller "sub-grams". The resulting visual n-gram models have substantial advantages over prior open-world visual models [125]: they recognize landmarks such as "Times Square", they differentiate between 'Washington DC" and the "Washington Nationals", and they distinguish between "city park" and "Park City".

Contribution. The technical contributions are threefold:

- We are the first to explore the prediction of n-grams relevant to image content using convolutional networks;
- We develop a novel, differentiable smoothing layers for such networks;
- We provide a simple solution to the out-of-vocabulary problem of traditional

image-recognition models.

We present a series of experiments to demonstrate the merits of our proposed model in image tagging, image retrieval, image captioning, and zero-shot transfer.

5.2 Related work

There is a substantial body of prior work that is related to this study, in particular, work on (1) learning from weakly supervised web data, (2) relating image content and language, and (3) language modeling. We give a (non-exhaustive) overview of prior work below.

5.2.1 Learning from weakly supervised web data

Several prior studies have used Google Images to obtain large collections of (weakly) labeled images for the training of vision models [122–124, 126–128, 132]. We do not opt for such an approach here because it is very difficult to understand the biases it introduces, in particular, because image retrieval by Google Images is likely aided by a content-based image retrieval model itself. This introduces the real danger that training on data from Google Images amounts to replicating an existing black-box vision system. Various other studies have used data from photo-sharing websites such as Flickr for training; for instance, to train hierarchical topic models [133] or multiple-instance learning SVMs [134], to learn label distribution models [135, 136], to finetune pretrained convolutional networks [137], and to train weak classifiers that produce additional visual features [138]. Like this study, [125]

trains convolutional networks on the image-comment pairs. Our study differs from [125] in that we do not just consider single words, as a result of which our models distinguish between, *e.g.*, "city park" and "Park City". Indeed, the models in [125] are a special case of our models in which only unigrams are considered.

5.2.2 Relating image content and language

Our approach is connected to a wide body of work that aims at bridging the semantic gap between vision and language [139]. In particular, many studies have explored this problem in the context of image captioning. Most image-captioning systems train a recurrent network or maximum entropy language model on top of object classifications produced by a convolutional network; the models are either trained separately [140–142] or end-to-end [143, 144]. We do not consider recurrent networks in our study because test-time inference in such networks is slow, which hampers the deployment of such models in real-world applications. An image-captioning study that is closely related to our work is [145], which trains a bilinear model that outputs phrase probabilities given an image feature and combines the relevant phrases into a caption using a collection of heuristics. Several other works have explored joint embedding of images and text, either at the word level [146] or at the sentence level [147, 148]. What distinguishes our study is that prior work is generally limited in the variety of visual concepts it can deal with; these studies rely on vision models that recognize only small numbers of classes and / or on the availability of "ground-truth" captions that describe the image content — such captions

are very different from a typical user comment on Flickr. In contrast to prior work, we consider the open-world setting with very large numbers of visual concepts, and we do not rely on ground-truth captions provided by human annotators. Our study is most similar to that of [149], which uses n-gram to generate image descriptions; unlike [149], we we do not rely on separately trained image-classification pipelines. Instead, we train our model end-to-end on a dataset without ground-truth labels.

5.2.3 Language models

Several prior studies have used phrase embeddings for natural language processing tasks such as named entity recognition [150], text classification [151–153], and machine translation [154, 155]. These studies differ from our work in that they focus solely on language modeling and not on visual recognition. Our models are inspired by smoothing techniques used in traditional n-gram language models[1], in particular, Jelinek-Mercer smoothing [130]. Our models differ from traditional n-gram language models in that they are *image-conditioned* and *parametric*: whereas n-gram models count the frequency of n-grams in a text corpus to produce a distribution over phrases or sentences, our model measures phrase likelihoods by evaluating inner products between image features and learned parameter vectors.

Below, we describe the dataset we use in our experiments, the loss functions we optimize, and the training procedure we use for optimization.

[1] A good overview of these techniques is given in [156, 157].

5.3 Dataset

We train our models on the YFCC100M dataset, which contains 99.2 million images and associated multi-lingual user comments [129]. We applied a simple language detector to the dataset to select only images with English user comments, leaving a total of 30 million examples for training and testing. We preprocessed the text by removing punctuations, and we added `[BEGIN]` and `[END]` tokens at the beginning and end of each sentence. We preprocess all images by rescaling them to 256×256 pixels (using bicubic interpolation), cropping the central 224×224, subtracting the mean pixel value of each image, and dividing by the standard deviation of the pixel values.

For most experiments, we use a dictionary of all English n-grams (with n between 1 and 5) with more than $1,000$ occurrences in the 30 million English comments. This dictionary contains $142,806$ n-grams: $22,869$ unigrams, $56,830$ bigrams, $32,560$ trigrams, $17,351$ four-grams, and $13,196$ five-grams. We emphasize that the smoothed visual n-gram models we describe below are trained and evaluated on all n-grams in the dataset, even if these n-grams are not in the dictionary. However, whereas the probability of in-dictionary n-grams is primarily a function of parameters that are specifically tuned for those n-grams, the probability of out-of-dictionary n-grams is composed from the probability of smaller in-dictionary n-grams (details below).

5.4 Loss functions

The main contribution is in the loss functions we use to train our phrase prediction models. In particular, we explore (1) a *naive n-gram loss* that measures the (negative) log-likelihood of in-dictionary n-grams that are present in a comment and (2) a *smoothed n-gram loss* that measures the (negative) log-likelihood of all n-grams, even if these n-grams are not in the dictionary. This loss uses smoothing to assign non-zero probabilities to out-of-dictionary n-grams; specifically, we experiment with Jelinek-Mercer smoothing [130].

5.4.1 Notation

We denote the input image by $\mathbf{I}$ and the image features extracted by the convolutional network with parameters θ by $\phi(\mathbf{I}; \theta) \in \mathbb{R}^D$. We denote the n-gram dictionary that our model uses by $\mathcal{D}$ and a comment containing K words by $w \in [1, C]^K$, where C is the total number of words in the (English) language. We denote the n-gram that ends at the i-th word of comment w by w_{i-n+1}^i and the i-th word in comment w by w_i^i. Our predictive distribution is governed by a n-gram embedding matrix $\mathbf{E} \in \mathbb{R}^{D \times |\mathcal{D}|}$. With a slight abuse of notation, we denote the embedding corresponding to a particular n-gram w by $\mathbf{e}_w$. For brevity, we omit the sum over all image-comment pairs in the training / test data when writing loss functions.

5.4.2 Naive n-gram loss

The naive n-gram loss is a standard multi-class logistic loss over all n-grams in the dictionary $\mathcal{D}$. The loss is summed over all n-grams that appear in the sentence w; that is, n-grams that do not appear in the dictionary are ignored:

$$\ell(\mathbf{I}, w; \theta, \mathbf{E}) = -\sum_{m=1}^{n} \sum_{i=n}^{K} \mathbb{I}\left[w_{i-n+1}^{i} \in \mathcal{D}\right] \log p_{obs}\left(w_{i-m+1}^{i} | \phi(\mathbf{I}; \theta); \mathbf{E}\right), \tag{5.1}$$

where the *observational likelihood* $p_{obs}(\cdot)$ is given by a softmax distribution over all in-dictionary n-grams that is governed by the inner product between the image features $\phi(\mathbf{I}; \theta)$ and the n-gram embeddings:

$$p_{obs}\left(w | \phi(\mathbf{I}; \theta); \mathbf{E}\right) = \frac{\exp\left(-\mathbf{e}_{w}^{\top} \phi(\mathbf{I}; \theta)\right)}{\sum_{w' \in \mathcal{D}} \exp\left(-\mathbf{e}_{w'}^{\top} \phi(\mathbf{I}; \theta)\right)}. \tag{5.2}$$

The image features $\phi(\mathbf{I}; \theta)$ are produced by a convolutional network $\phi(\cdot)$, which we describe in more detail in 5.5.

The naive n-gram loss cannot do language modeling because it does not model a conditional probability. To circumvent this issue, we construct an ad-hoc conditional distribution based on the scores produced by our model at prediction time using a "stupid" back-off model [158]:

$$p\left(w_i^i | w_{i-n+1}^{i-1}\right) = \begin{cases} p_{obs}\left(w_i^i | w_{i-n+1}^{i-1}\right), & \text{if } w_{i-n+1}^{i} \in \mathcal{D} \\ \lambda p\left(w_i^i | w_{i-n+2}^{i-1}\right), & \text{otherwise.} \end{cases} \tag{5.3}$$

For brevity, we dropped the conditioning on $\phi(\mathbf{I}; \theta)$ and $\mathbf{E}$.

5.4.3 Jelinek-Mercer (J-M) loss

The simple n-gram loss has two main disadvantages: (1) it ignores out-of-dictionary n-grams entirely during training and (2) the parameters $\mathbf{E}$ that correspond to infrequent in-dictionary words are difficult to pin down. The Jelinek-Mercer loss aims to address both these issues. The loss is defined as:

$$\ell(\mathbf{I}, w; \theta, \mathbf{E}) = -\sum_{i=1}^{K} \log p\left(w_i^i | w_{i-n+1}^{i-1}, \phi(\mathbf{I}; \theta); \mathbf{E}\right), \tag{5.4}$$

where the likelihood of a word conditioned on the $(n-1)$ words appearing before it is defined as:

$$p\left(w_i^i | w_{i-n+1}^{i-1}\right) = \lambda p_{obs}\left(w_i^i | w_{i-n+1}^{i-1}\right) + (1-\lambda) p\left(w_i^i | w_{i-n+2}^{i-1}\right). \tag{5.5}$$

Herein, we removed the conditioning on $\phi(\mathbf{I}; \theta)$ and $\mathbf{E}$ for brevity. The parameter $0 \leq \lambda \leq 1$ is a smoothing constant that governs how much of the probability mass from $(n-1)$-grams is (recursively) transferred to both in-dictionary and out-of-dictionary n-grams. The probability mass transfer prevents the Jelinek-Mercer loss from assigning zero probability (which would lead to infinite loss) to out-of-vocabulary n-grams, and it allows it to learn from low-frequency and out-of-vocabulary n-grams.

The Jelinek-Mercer loss is differentiable with respect to both $\mathbf{E}$ and θ, as a result of which the loss can be backpropagated through the convolutional network. In particular, the loss gradient with respect to ϕ is given by:

$$\frac{\partial \ell}{\partial \phi} = -\sum_{i=1}^{K} p\left(w_i^i | w_{i-n+1}^{i-1}, \phi(\mathbf{I}; \theta); \mathbf{E}\right) \frac{\partial p}{\partial \phi}, \tag{5.6}$$

where the partial derivatives are given by:

$$\frac{\partial p}{\partial \phi} = \lambda \frac{\partial p_{obs}}{\partial \phi} + (1 - \lambda) \frac{\partial p}{\partial \phi} \tag{5.7}$$

$$\frac{\partial p_{obs}}{\partial \phi} = p_{obs}(w|\phi(\mathbf{I};\theta);\mathbf{E}) \left(\mathbb{E}[\mathbf{e}_{w'}]_{w' \sim p_{obs}} - \mathbf{e}_w \right). \tag{5.8}$$

This error signal can be backpropagated directly through the convolutional network $\phi(\cdot)$.

5.5 Training

The core of our visual recognition models is formed by a convolutional network $\phi(\mathbf{I};\theta)$. For expediency, we opt for a residual network [32] with 34 layers. Our networks are initialized by an Imagenet-trained network, and trained to minimize the loss functions described above using stochastic gradient descent using a batch size of 128 for 10 epochs. In all experiments, we employ Nesterov momentum of 0.9, a weight decay of 0.0001, and an initial learning rate of 0.1; the learning rate is divided by 10 whenever the training loss stabilizes (until a minimum learning rate of 0.001).

A major bottleneck in training is the large number of outputs of our observation model: doing a forward-backward pass with 512 inputs (the image features) and $142,806$ outputs (the n-grams) is computationally intensive. To circumvent this issue, we follow [125] and perform *stochastic gradient descent over outputs* [159]: we only perform the forward-backward pass for a random subset (formed by all positive n-grams in the batch) of the columns of $\mathbf{E}$. This simple approximation works well in practice, and it can be shown to be closely related to the exact loss [125].

5.6 Experiments

Below, we present the four sets of experiments we performed to assess the performance of our visual n-gram models in: (1) phrase-level image tagging, (2) phrase-based image retrieval, (3) relating images and captions, and (4) zero-shot transfer.

5.6.1 Phrase-level image tagging

We first gauge whether relevant comments for images have high likelihood under our visual n-gram models. Specifically, we measure the average perplexity of predicting the correct words in a comment on a held-out test set of $10,000$ images. We only consider in-dictionary n-grams in our perplexity measurements. The perplexity of a model is defined as $2^{H(p)}$, where $H(p)$ is the cross-entropy:

$$H(p) = -\frac{1}{K}\sum_{i=1}^{K} \log_2 p\left(w_i^i | w_{i-n+1}^{i-1}, \phi(\mathbf{I};\theta); \mathbf{E}\right). \tag{5.9}$$

Based on the results of preliminary experiments on a held-out validation set, we set $\lambda = 0.2$ in the Jelinek-Mercer loss. Following common practice in language modeling [157], we replace the likelihood of out-of-vocabulary unigrams in the results for naive n-gram loss by a uniform distribution over unigrams: this prevents the perplexity from becoming infinite.

We compare models that use either of the two loss functions (the naive in-dictionary n-gram loss and Jelinek-Mercer loss) with a baseline trained with a linear layer on top of Imagenet-trained visual features trained using naive n-gram loss. We

Table 5.1: Perplexity of visual n-gram models on in-dictionary n-grams, measured on YFCC100M test set of $10,000$ images comprising $268,972$ five-grams. Results for two losses (rows) with and without smoothing at test time (columns). Lower is better.

Loss / Smoothing	**"Stupid" back-off**	**Jelinek-Mercer**
Imagenet + linear	349	233
Naive n-gram	297	212
Jelinek-Mercer	276	**199**

consider two settings of our models at prediction time: (1) a setting in which we use the "stupid" back-off model with $\lambda = 1$; and (2) a setting in which we smooth the $p(\cdot)$ predictions using Jelinek-Mercer smoothing (as described above).

The resulting perplexities for all experimental settings are presented in Table 5.1. From the results presented in the table, we observe that: (1) the use of smoothing losses for training image-based phrase prediction models leads to better models than the use of a naive n-gram loss; and (2) the use of additional smoothing at test time may further reduce the perplexity of the n-gram model. The former effect is the result of the ability of smoothing losses to direct the learning signal to the most relevant n-grams instead of equally spreading it over all n-grams that are present in the target. The latter effect is the result of the ability of prediction-time smoothing to propagate the probability mass from in-dictionary n-grams to relevant out-of-dictionary n-grams.

To obtain more insight into the phrase-prediction performance of our models, we also assess our model's ability to predict relevant phrases (n-grams) for images. To correct for variations in the marginal frequency of n-grams, we calibrate all log-likelihood scores by subtracting the average log-likelihood our model predicts

Table 5.2: Phrase-prediction performance on YFCC100M test set of 10,000 images measured in terms of recall@k at three cut-off levels k (lefthand-side; see text for details) and the percentage of correctly predicted n-grams according to human raters (righthand-side) for one baseline model and two of our phrase prediction models. Higher is better.

Model	**R@1**	**R@5**	**R@10**	**Accuracy**
Imagenet + linear	5.0	10.7	14.5	32.7
Naive n-gram	5.5	11.6	15.1	36.4
Jelinek-Mercer	**6.2**	**13.0**	**18.1**	**42.0**

on a large collection of held-out validation images. We predict n-gram phrases for images by outputting the n-grams with the highest calibrated log-likelihood score for an image. Examples of the resulting n-gram predictions are shown in Figure 5.1.

We quantify phrase-prediction performance in terms of recall@k on a set of 10,000 images from the YFCC100M test set. We define recall@k as the average percentage of n-grams appearing in the comment that are among the k front-ranked n-grams when the n-grams are sorted according to their score under the model. In this experiment and all experiments hereafter, we only present results where the same smoothing is used at training and at prediction time: that is, we use the "stupid" back-off model on the predictions of naive n-grams models and we smooth the predictions of Jelinek-Mercer models using Jelinek-Mercer smoothing. As a baseline, we consider a linear multi-class classifier over n-grams (*i.e.*, using naive n-gram loss) trained on features produced by an Imagenet-trained convolutional network. The results are shown in the lefthand-side of Table 5.2.

Because the n-grams in the YFCC100M test set are noisy targets (many words that are relevant to the image content are not present in the comments), we also

performed an experiment on Amazon Mechanical Turk in which we asked two human raters whether or not the highest-scoring n-gram was relevant to the content of the image. We filter out unreliable raters based on their response time, and for each of our models, we measure the percentage of retrieved n-grams that is considered relevant by the remaining raters. The resulting accuracies of the visual n-gram models are reported in the righthand-side of Table 5.2.

The results presented in the table are in line with the results presented in Table 5.1: they show that the use of a smoothing loss substantially improves the results compared to baseline models based on the naive n-gram loss. In particular, the relative performance in recall@k between our best model and the Imagenet-trained baseline model is approximately 20%. The merits of the Jelinek-Mercer loss are confirmed by our experiment on Mechanical Turk: according to human annotators, 42.0% of the predicted phrases is relevant to the visual content of the image.

Next, we study the performance of our Jelinek-Mercer model as a function of n; that is, we investigate the effect of including longer n-grams in our model on the model performance. As before, we measure recall@k of n-gram retrieval as a function of the cut-off level k, and consider models with unigrams to five-grams. Figure 5.2 presents the results of this experiment, which shows that the performance of our models increases as we include longer n-grams in the dictionary. The figure also reveals diminishing returns: the improvements obtained from going beyond trigrams are limited.

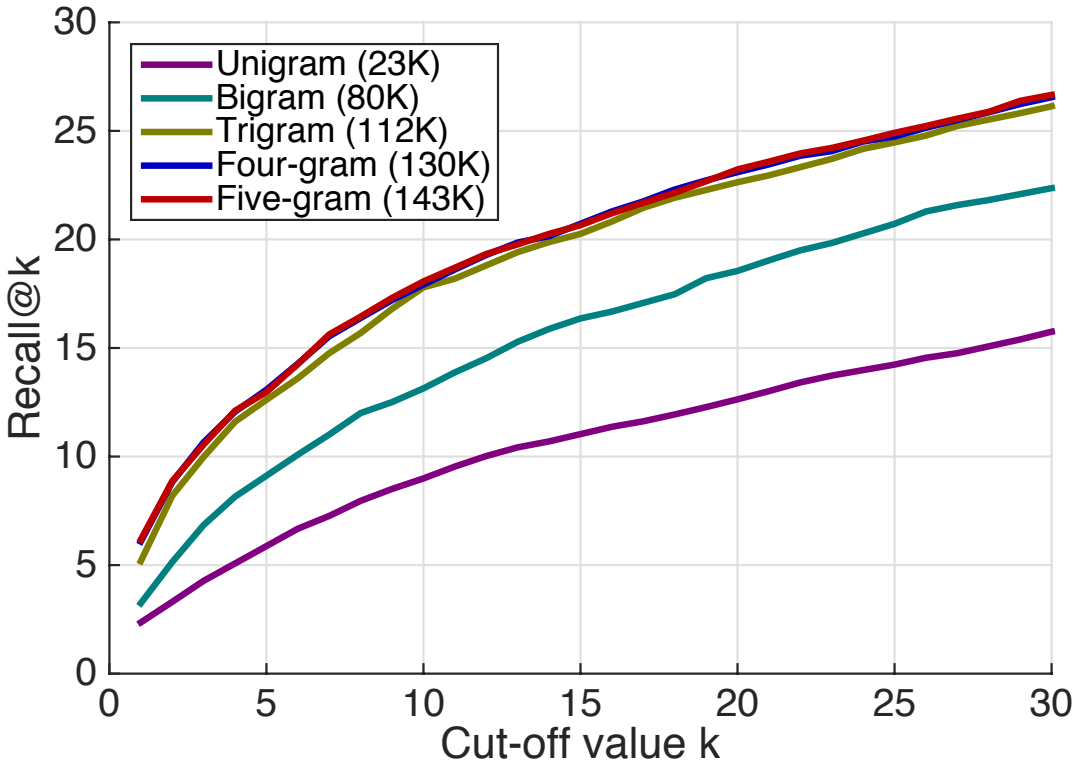


Figure 5.2: Recall@k on n-gram retrieval of five models with increasing maximum length of n-grams included in the dictionary ($n = 1, \ldots, 5$), for varying cut-off values k. The dictionary size of each of the models is shown between brackets. Higher is better.

5.6.2 Phrase-based image retrieval

In the second set of experiments, we measure the ability of the system to retrieve relevant images for a given n-gram query. Specifically, we rank all images in the test set according to the calibrated log-likelihood our models predict for the query-image pairs.

In Figure 5.3, we show examples of twelve images that are most relevant from a set of $931,588$ YFCC100M test images (according to our model) for four different n-gram queries; we manually picked these n-grams to demonstrate the merits of building phrase-level image recognition models. The figure shows that the model has learned accurate visual representations for n-grams such as "Market Street" and "street market", as well as for "city park" and "Park City" (see the caption of Figure 5.3 for details on the queries). We show a second set of image retrieval examples in Figure 5.4, which shows that our model is able to distinguish visual concepts related to Washington: namely, between the state, the city, the baseball team, and the hockey team.

As in our earlier experiments, we quantify the image-retrieval quality of our model on a set of $10,000$ test images from the YFCC100M dataset by measuring the precision and recall of retrieving the correct image given a query n-grams. We compute a precision-recall curve by averaging over the $10,000$ n-gram queries that have the highest tf-idf value in the YFCC100M dataset: the resulting curve is shown in Figure 5.5. The results from this experiment are in accordance with the previous results: the naive n-gram loss substantially outperforms our Imagenet baseline,

Figure 5.3: Four highest-scoring images for n-gram queries "Market Street", "street market", "city park", and "Park City" from a collection of 931,588 YFCC100M images. Market Street is a common street name, for instance, it is one of the main thoroughfares in San Francisco. Park City (Utah) is a popular winter sport destination. The figure only shows images from the YFCC100M dataset whose license allows reproduction. We refer to the appendix for detailed copyright information.

Table 5.3: Caption retrieval performance on YFCC100M test set of $10,000$ images measured in terms of recall@k at three cut-off levels k (lefthand-side; see text for details) and the percentage of correctly retrieved captions according to human raters (righthand-side) one baseline model and two of our phrase prediction models. Higher is better.

Model	**R@1**	**R@5**	**R@10**	**Accuracy**
Imagenet + linear	1.1	3.3	4.8	38.3
Naive n-gram	1.3	4.4	6.9	42.0
Jelinek-Mercer	**7.1**	**16.7**	**21.5**	**53.1**

which in turn, is outperformed by the model trained using Jelinek-Mercer loss. Admittedly, the precisions we obtain are fairly low even in the low-recall regime. This low recall is the result of the false-negative noise in the "ground truth" we use for evaluation: an image that is relevant to the n-gram query may not be associated with that n-gram in the YFCC100M dataset, as a result of which we may consider it as "incorrect" even when it ought to be correct based on the visual content of the image.

5.6.3 Relating Images and Captions

In the third set of experiments, we study to whether visual n-gram models can be used for relating images and captions. While many image-conditioned language models have focused on caption generation, accurately measuring the quality of a model is still an open problem: most current metrics poor correlated with human judgement [165]. Therefore, we focus on caption-based retrieval tasks instead: in particular, we evaluate the performance of our models in caption-based image retrieval and image-based caption retrieval. In caption-based image retrieval, we

Figure 5.4: Four highest-scoring images for n-gram queries "Washington State", "Washington DC", "Washington Nationals", and "Washington Capitals" from a collection of 931,588 YFCC100M test images. Washington Nationals is a Major League Baseball team; Washington Capitals is a National Hockey League hockey team. The figure only shows images from the YFCC100M dataset whose license allows reproduction. We refer to the appendix for detailed copyright information.

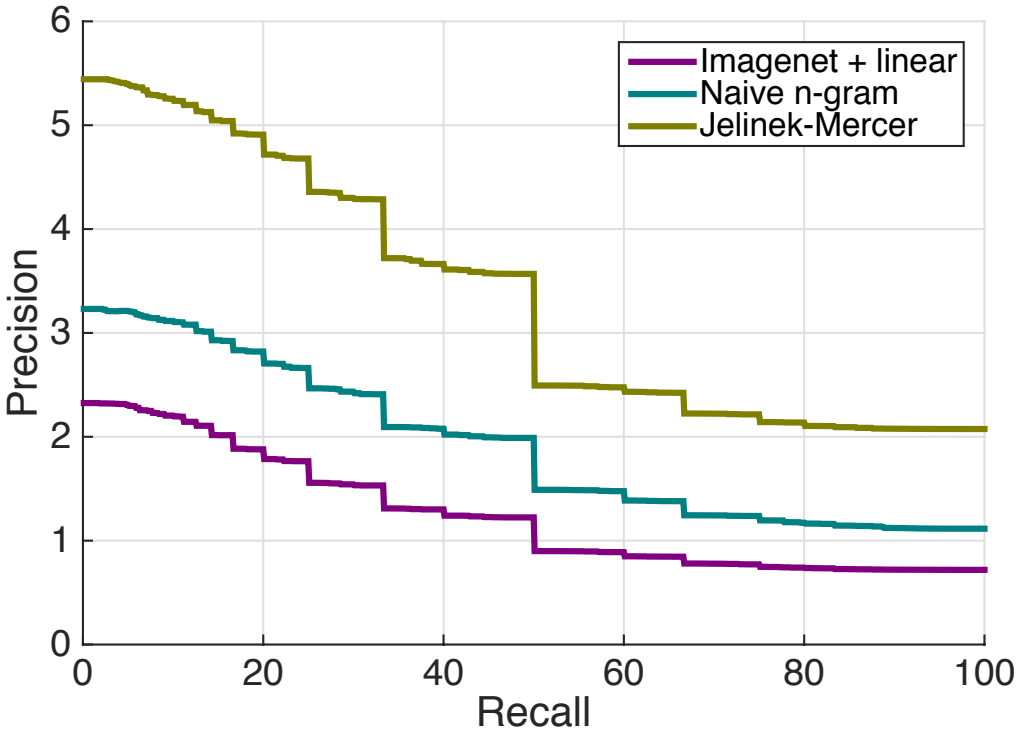


Figure 5.5: Precision-recall curve for phrase-based image retrieval of our models on YFCC100M test set of $10,000$ images one baseline model and two of our phrase-prediction models. The curves were obtained by averaging over the $10,000$ n-gram queries with the highest tf-idf value.

rank images according to their log-likelihood for a particular caption and measure recall@k: the percentage of queries for which the correct image is among the k first images.

We first perform an experiment on $10,000$ images and comments from the YFCC100M test set. In addition to recall@k, we also measure accuracy by asking two human raters to assess whether the retrieved caption is relevant to the image content. The results of these experiments are presented in Table 5.3: they show that the strong performance of our visual n-gram models extends to caption retrieval[2].

[2]We also performed experiments with a neural image captioning model that was trained on

Table 5.4: Recall@k (for three cut-off levels k) of caption-based image retrieval on the COCO-5K and Flickr-30K datasets for eight baseline models and our visual n-gram models (with and without finetuning). Baselines are separated in models dedicated to retrieval (top) and image-conditioned language models (bottom).

Image retrieval	**COCO-5K**			**Flickr-30K**		
	R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
Retrieval models						
Karpathy et al. [148]	–	–	–	10.2	30.8	44.2
Klein et al. [160]	**11.2**	**29.2**	**41.0**	25.0	52.7	66.0
Deep CCA [161]	–	–	–	26.8	52.9	66.9
Wang et al. [162]	–	–	–	**29.7**	**60.1**	**72.1**
Language models						
STD-RNN [163]	–	–	–	8.9	29.8	41.1
BRNN [141]	10.7	29.6	42.2	15.2	37.7	50.5
Kiros et al. [164]	–	–	–	16.8	42.0	56.5
NIC [144]	–	–	–	17.0	–	57.0
Ours						
Naive n-gram	0.3	1.1	2.1	1.0	2.9	4.9
Jelinek-Mercer	5.0	14.5	21.9	8.8	21.2	29.9
J-M + finetuning	11.0	29.0	40.2	17.6	39.4	50.8

According to human raters, our best model retrieves a relevant caption for 53.1% of the images in the test set. To assess if visual n-grams help, we also experiment with a unigram model [125] with a dictionary size of 142,806. We find that this model performs worse than visual n-gram models: its recall@k scores of are 1.2, 4.2, and 6.3, respectively.

To facilitate comparison with existing methods, we also perform experiments on the COCO-5K and Flickr-30K datasets [120, 166] using visual n-gram models

COCO [144], but this model performs poorly: it obtains a recall@k of 0.2, 1.0, and 1.6 for $k=1$, 5, and 10, respectively. This is because many of the words that appear in YFCC100M are not in COCO.

trained on YFCC100M[3]. The results of these experiments are presented in Table 5.4; they show that our model performs roughly on par with the state-of-the-art based on language models on both datasets. We emphasize that our models have much larger vocabularies than the baseline models, which implies the strong performance of our models likely generalizes to a much larger visual vocabulary than the vocabulary required to perform well on COCO-5K and Flickr-30K. Like other language models, our models perform worse on the Flickr-30K dataset than dedicated retrieval models [148, 160–162]. Interestingly, our model does perform on par with a state-of-the-art retrieval model [160] on COCO-5K.

We also perform image-based caption retrieval experiments: we retrieve captions by ranking all captions in the COCO-5K and Flick-30K test set according to their log-likelihood under our model. The results of this experiment are presented in Table 5.5, which shows that our model performs on par with state-of-the-art image-conditioned language models on caption retrieval. Like all other language models, our model performs worse than approaches tailored towards retrieval on the Flickr-30K dataset. On COCO-5K, visual n-grams perform on par with the state-of-the-art.

5.6.4 Zero-Shot Transfer

Because our models are trained on approximately 30 million photos and comments, they have learned to recognize a wide variety of visual concepts. To assess

[3]Please refer to Appendix C for additional results in the COCO-1K dataset and additional baseline models for relating images and captions.

Table 5.5: Recall@k (for three cut-off levels k) of caption retrieval on the COCO-5K and Flickr-30K datasets for eight baseline systems and our visual n-gram models (with and without finetuning). Baselines are separated in models dedicated to retrieval (top) and image-conditioned language models (bottom). Higher is better.

Caption retrieval	**COCO-5K**			**Flickr-30K**		
	R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
Retrieval models						
Karpathy et al. [148]	–	–	–	16.4	40.2	54.7
Klein et al. [160]	17.7	40.1	51.9	35.0	62.0	73.8
Deep CCA [161]	–	–	–	27.9	56.9	68.2
Wang et al. [162]	–	–	–	**40.3**	**68.9**	**79.9**
Language models						
STD-RNN [163]	–	–	–	9.6	29.8	41.1
BRNN [141]	16.5	39.2	52.0	22.2	48.2	61.4
Kiros et al. [164]	–	–	–	23.0	50.7	62.9
NIC [144]	–	–	–	23.0	–	63.0
Ours						
Naive n-gram	0.7	2.8	4.6	1.2	5.9	9.6
Jelinek-Mercer	8.7	23.1	33.3	15.4	35.7	45.1
J-M + finetuning	**17.8**	**41.9**	**53.9**	28.6	54.7	66.0

the ability of our models to recognize visual concepts out-of-the-box, we perform a series of *zero-shot transfer* experiments. Unlike traditional zero-shot learners (*e.g.*, [167–169]), we simply apply the Flickr-trained models on a test set from a different dataset. We automatically match the classes in the target dataset with the n-grams in our dictionary. We perform experiments on the aYahoo dataset [170], the SUN dataset [171], and the Imagenet dataset [28]. For a test image, we rank the classes that appear in each dataset according to the score our model assigns to the corresponding n-grams, and predict the highest-scoring class for that image. We report the accuracy of the resulting classifier in Table 5.6 in two settings: (1) a setting in which performance is measured only on in-dictionary classes and (2) a

Table 5.6: Classification accuracies on three zero-shot transfer learning datasets on in-dictionary and on all classes. The number of in-dictionary classes is 10 out of 12 for aYahoo, 326 out of 1,000 for Imagenet, and 330 out of 720 for SUN. Higher is better.

	aYahoo	**Imagenet**	**SUN**
Class mode (in dictionary)	15.3	0.3	13.0
Class mode (all classes)	12.5	0.1	8.6
Jelinek-Mercer (in dictionary)	88.9	35.2	34.7
Jelinek-Mercer (all classes)	72.4	11.5	23.0

setting in which performance is measured on all classes.

The results of these experiments are shown in Table 5.6. For reference, we also present the performance of a model that always predicts the a-priori most likely class. The results reveal that, even without any finetuning or re-calibration, non-trivial performances can be obtained on generic vision tasks. The performance of our models is particularly good on common classes such as those in the aYahoo dataset for which many examples are available in the YFCC100M dataset. The performance of our models is worse on datasets that involve fine-grained classification such as Imagenet, for instance, because YFCC100M contains few examples of specific, uncommon dog breeds.

5.7 Discussions

5.7.1 Visual n-grams and recurrent models

This study has presented a simple yet viable alternative to the common practice of training a combination of convolutional and recurrent networks to relate

images and language. Our visual n-gram models differ in several key aspects from models based on recurrent networks. Visual n-gram models are less suitable for caption generation[4] [173] but they are much more efficient to evaluate at inference time, which is very important in real-world applications of these models. Visual n-grams are more interpretable than recurrent models because the likelihood of any n-gram or sentence can be readily evaluated and ranked. This allows to compute test log-likelihoods for entire sentences, instead of just the log-likelihood of a single, subsequent word. Such test log-likelihoods can, in turn, be used to perform (Bayesian) model comparison.

Visual n-gram models can be combined with class activation mapping [174,175] to perform visual grounding of n-grams, as shown in Figure 5.6. Such grounding is facilitated by the close relation between predicting visual n-grams and standard image classification. This makes visual n-gram models more amenable to transfer to new tasks than approaches based on recurrent models, as demonstrated by our zero-shot transfer experiments.

Whilst a recurrent model trained on the YFCC100M dataset can only be used for image captioning, visual n-grams models trained on the same dataset can be used in a range of tasks, including image tagging, image retrieval, image captioning, class discovery, and traditional image classification.

[4] Our model achieves a METEOR score [172] of 17.2 on COCO captioning with a test set of $1,000$ images, versus 15.7 for a nearest neighbor baseline method and 19.5 for a recurrent network [141].

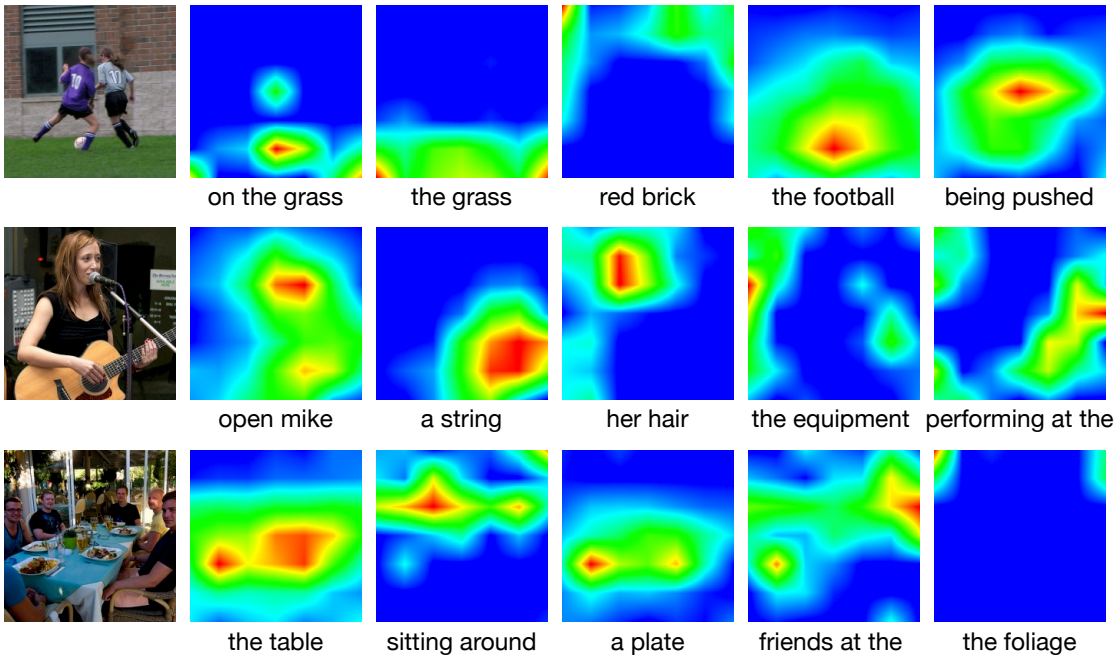


Figure 5.6: Discriminative regions of five n-grams for three images, computed using class activation mapping [174, 175].

5.7.2 Learning from web data

Another important aspect that discerns our work from most work in computer vision is that our models are capable of being learned purely from web data, *without any manual data annotation*. We believe that this type of training is essential if we want to construct models that are not limited to a small visual vocabulary and that are readily applicable to real-world computer-vision tasks. Indeed, this paper fits in a recent line of work [123, 125] that abandons the traditional approach of gathering images, manually annotating them for a small visual vocabulary, and training and testing on the resulting image-target distribution. As a result, models such as ours may not necessarily achieve state-of-the-art results on established benchmarks, be-

cause they did not learn to exploit the biases of those benchmarks as well [176–178]. Such "negative" results highlight the necessity of developing less biased benchmarks that provide more signal on progress towards visual understanding.

Such approaches suffer from strong biases that do not reflect biases in the real world [179] and do not scale to the large visual vocabularies that are necessary for vision systems to operate in open-world setting. Moreover, models trained on datasets such as the YFCC100M dataset tend to be less biased towards the annotations that computer-vision researchers assign to their image data, but instead, possess biases that are more in line with the distribution by which the real world generates images. As a results, models such as those studied in this paper may not always achieve state-of-the-art results on established benchmarks, simply because they did not learn the biases of those benchmarks well. When interpreting such seemingly negative results, it is thus essential to continuously reflect on to what extent benchmarks provide signal on whether we are progressing towards the hallmark of true visual understanding.

For the reasons outlined above, we believe it will become increasingly common to train vision models on web data, and only use image annotation to analyze the capabilities of the resulting systems. In this paper, we have adopted this approach by (1) asking human raters to assess the relevance of captions or phrases to images and (2) by transferring our models to benchmark datasets for image captioning and attribute recognition.

5.8 Future work

The Jelinek-Mercer loss we studied in this paper is based on just one of many n-gram smoothers [157]. An interesting future work is to perform an in-depth comparison of different smoothers for the training of convolutional networks. In particular, we will consider loss functions based as absolute-discounting smoothing such as Kneser-Ney smoothing [131], as well as back-off models [180].

$$p\left(w_i^i|w_{i-n+1}^{i-1},\phi(\mathbf{I};\theta);\mathbf{E}\right)=\gamma_{i-n+1}^{i-1}p\left(w_i^i|w_{i-n+2}^{i-1},\phi(\mathbf{I};\theta);\mathbf{E}\right)$$
$$+\max\left\{p_{obs}\left(w_i^i|w_{i-n+1}^{i-1},\phi(\mathbf{I};\theta);\mathbf{E}\right)-\delta,0\right\}.$$

Herein, δ is an absolute discount factor that governs how much probability mass from $(n-1)$-grams is transferred to n-grams, and γ_{i-n+1}^{i-1} is a term that ensures the likelihood function remains a probability distribution (that is, that it sums up to one). Specifically, γ_{i-n+1}^{i-1} is given by:

$$\gamma_{i-n+1}^{i-1}=\sum_{w'}\min\left\{p_{obs}\left(w'|w_{i-n+1}^{i-1},\phi(\mathbf{I};\theta);\mathbf{E}\right)\right),\delta\right\}.$$

Another direction of future research is to explore the use of visual n-gram models in systems that operate in open-world settings, combining them with techniques for zero-shot and few-shot learning. Finally, it would be interesting to see how well our models could be applied to tasks that require recognition of a large variety of visual concepts and relations between them, such as visual question answering [181, 182], visual Turing tests [183], and scene graph prediction [184].

5.9 Conclusion

We presented a way to learn end-to-end visual n-gram models from large scale weakly supervised data. We proposed a novel smoothing loss function that utilizes the relationships between higher-order n-gram labels with lower-order n-grams and handles the out-of-vocabulary problem in traditional models, called Jelinek-Mercer loss. The visual n-gram models are applied and evaluated in multiple tasks such as predicting phrases from images, retrieving images from phrase queries, retrieving captions from images, retrieving images from captions and transfer learning. These experiments demonstrated the effectiveness of the proposed Jelinek-Mercer loss function. The most interesting part of our model is probably the capability of capturing the visual patterns for an extremely large number of n-gram concepts. It is mostly due to the fact that the proposed model can be efficiently trained from millions of paired images and user comments.

Chapter 6: Conclusion

We investigated ways to improve the robustness, interpretability and scalability of visual representations in four different works. These works are motivated by real world applications such as image-based geolocation, active face authentication, text-baesd image retrieval and phrase prediction from images. The works also studies geometric representations, low-level feature encoding, mid-level representation and high-level representation learning, respectively.

We show in our works that (1) incorporating feature space uncertainty estimation in the visual represenation can significantly improve the effectiveness and robustness of the features, (2) mid-level abstract representations can be effective in linking multimodal data, and (3) learning from large scale weakly supervised data is a promising way for representation learning.

While computer vision has started to work in real applications, there are still many problems remaining unsolved for building robust, interpretable and scalable visual representations. These include but not limited to learning representations from noisy, biased and small data, increasing the transferability of representations, incorporating uncertainty modeling in complex models for representation learning and incorporating generative models of mid-level concepts in representations.

Appendix A: Proofs in the projective uncertainty of line segments

A.1 Uncertainty modeling

Proof of Lemma 2.1.

$$\int_{t_1}^{t_2} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\|\mathbf{a}t+\mathbf{b}\|^2}{2\sigma^2}} dt \tag{A.1}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{t_1}^{t_2} e^{-\frac{\|\mathbf{a}\|^2 t^2 + 2\mathbf{a}^\top \mathbf{b} t + \|\mathbf{b}\|^2}{2\sigma^2}} dt \tag{A.2}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{t_1}^{t_2} e^{-\frac{\left(\|\mathbf{a}\| t + \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|}\right)^2 - \frac{(\mathbf{a}^\top \mathbf{b})^2}{\|\mathbf{a}\|^2} + \|\mathbf{b}\|^2}{2\sigma^2}} dt \tag{A.3}$$

$$= e^{-\frac{\|\mathbf{a}\|^2\|\mathbf{b}\|^2 - (\mathbf{a}^\top \mathbf{b})^2}{2\sigma^2\|\mathbf{a}\|^2}} \int_{t_1}^{t_2} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(t + \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|^2}\right)^2}{2\sigma^2/\|\mathbf{a}\|^2}} dt \tag{A.4}$$

$$= e^{-\frac{\|\mathbf{a}\|^2\|\mathbf{b}\|^2 - (\mathbf{a}^\top \mathbf{b})^2}{2\sigma^2\|\mathbf{a}\|^2}} \cdot \frac{1}{2} \left(\operatorname{erf}\left(\frac{t_2 + \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|^2}}{\sqrt{2}\sigma/\|\mathbf{a}\|} \right) - \operatorname{erf}\left(\frac{t_1 + \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|^2}}{\sqrt{2}\sigma/\|\mathbf{a}\|} \right) \right) \tag{A.5}$$

Since $\|\mathbf{a}\| = 1$, hence

$$\int_{t_1}^{t_2} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\|\mathbf{a}t+\mathbf{b}\|^2}{2\sigma^2}} dt$$

$$= e^{-\frac{\|\mathbf{b}\|^2 - (\mathbf{a}^\top \mathbf{b})^2}{2\sigma^2}} \cdot \frac{1}{2} \left(\operatorname{erf}\left(\frac{t_2 + \mathbf{a}^\top \mathbf{b}}{\sqrt{2}\sigma} \right) - \operatorname{erf}\left(\frac{t_1 + \mathbf{a}^\top \mathbf{b}}{\sqrt{2}\sigma} \right) \right) \tag{A.6}$$

□

Proof of Theorem 2.1. Let $p_n(\mathbf{x}; \boldsymbol{\mu}, \sigma^2)$ be the probability density function for nor-

mal distribution $N(\boldsymbol{\mu}, \sigma^2)$, i.e.

$$p_n(\mathbf{x}; \boldsymbol{\mu}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\|\mathbf{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} \tag{A.7}$$

The probability that $\mathbf{x}$ lies on the line segment equals the probability that random variables of the two ending points are $\mathbf{x} + t_a\Delta_\varphi$ and $\mathbf{x} + t_b\Delta_\varphi$ for some $t_a, t_b \in \mathbb{R}$ and $t_a \cdot t_b \leq 0$, therefore

$$\begin{aligned} p(\mathbf{x}, \varphi|\mathbf{a}, \mathbf{b}) = &\int_{-\infty}^{0} p_n(\mathbf{x} + t\Delta_\varphi; \mathbf{a}, \sigma^2)dt \int_{0}^{\infty} p_n(\mathbf{x} + t\Delta_\varphi; \mathbf{b}, \sigma^2)dt \\ &+ \int_{0}^{\infty} p_n(\mathbf{x} + t\Delta_\varphi; \mathbf{a}, \sigma^2)dt \int_{-\infty}^{0} p_n(\mathbf{x} + t\Delta_\varphi; \mathbf{b}, \sigma^2)dt \end{aligned} \tag{A.8}$$

According to Lemma 2.1,

$$\int_{-\infty}^{0} p_n(\mathbf{x} + t\Delta_\varphi; \mathbf{a}, \sigma^2)dt \int_{0}^{\infty} p_n(\mathbf{x} + t\Delta_\varphi; \mathbf{b}, \sigma^2)dt \tag{A.9}$$

$$= \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\|\Delta_\varphi t + \mathbf{x} - \mathbf{a}\|^2}{2\sigma^2}} dt \int_{0}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\|\Delta_\varphi t + \mathbf{x} - \mathbf{b}\|^2}{2\sigma^2}} dt \tag{A.10}$$

$$\begin{aligned} &= e^{-\frac{\|\mathbf{x}-\mathbf{a}\|^2 - |\langle\Delta_\varphi, \mathbf{x}-\mathbf{a}\rangle|^2}{2\sigma^2}} \cdot \frac{1}{2}\left(\operatorname{erf}\left(\frac{\langle\Delta_\varphi, \mathbf{x}-\mathbf{a}\rangle}{\sqrt{2}\sigma}\right) + 1\right) \\ &\cdot e^{-\frac{\|\mathbf{x}-\mathbf{a}\|^2 - |\langle\Delta_\varphi, \mathbf{x}-\mathbf{b}\rangle|^2}{2\sigma^2}} \cdot \frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{\langle\Delta_\varphi, \mathbf{x}-\mathbf{b}\rangle}{\sqrt{2}\sigma}\right)\right) \end{aligned} \tag{A.11}$$

$$= \frac{1}{4}C\left(1 + \operatorname{erf}\left(\frac{\langle\Delta_\varphi, \mathbf{x}-\mathbf{a}\rangle}{\sqrt{2}\sigma}\right)\right)\left(1 - \operatorname{erf}\left(\frac{\langle\Delta_\varphi, \mathbf{x}-\mathbf{b}\rangle}{\sqrt{2}\sigma}\right)\right) \tag{A.12}$$

where

$$C = \exp\left(-\frac{\|\mathbf{x}-\mathbf{a}\|^2 - |\langle\Delta_\varphi, \mathbf{x}-\mathbf{a}\rangle|^2 + \|\mathbf{x}-\mathbf{a}\|^2 - |\langle\Delta_\varphi, \mathbf{x}-\mathbf{b}\rangle|^2}{2\sigma^2}\right). \tag{A.13}$$

Similarly, it can also be derived that

$$\begin{aligned} &\int_{0}^{\infty} p_n(\mathbf{x} + t\Delta_\varphi; \mathbf{a}, \sigma^2)dt \int_{-\infty}^{0} p_n(\mathbf{x} + t\Delta_\varphi; \mathbf{b}, \sigma^2)dt \\ &= \frac{1}{4}C\left(1 - \operatorname{erf}\left(\frac{\langle\Delta_\varphi, \mathbf{x}-\mathbf{a}\rangle}{\sqrt{2}\sigma}\right)\right)\left(1 + \operatorname{erf}\left(\frac{\langle\Delta_\varphi, \mathbf{x}-\mathbf{b}\rangle}{\sqrt{2}\sigma}\right)\right) \end{aligned} \tag{A.14}$$

Therefore,

$$p(\mathbf{x}, \varphi|\mathbf{a}, \mathbf{b}) = \frac{1}{2}C\left(1 - \operatorname{erf}\left(\frac{\langle \mathbf{x}-\mathbf{a}, \Delta_\varphi\rangle}{\sqrt{2}\sigma}\right)\operatorname{erf}\left(\frac{\langle \mathbf{x}-\mathbf{b}, \Delta_\varphi\rangle}{\sqrt{2}\sigma}\right)\right) \tag{A.15}$$

which is equivalent to Equation 2.3. □

A.2 Line slopes under projective transformation

The point coordinate transformed by $\mathbf{Q}$ can be obtained by homogeneous coordinate representation. For the slope angle, let $\mathbf{q}_i$ be the i-th row vector of projection matrix $\mathbf{Q}$, the transformed slope angle at location $\mathbf{x} = (x, y)^\top$ is φ'. Then $\tan\varphi' = \frac{dy'}{dx'}$ where

$$x'_{(x,y)} = \frac{\mathbf{q}_1^\top(x, y, 1)^\top}{\mathbf{q}_3^\top(x, y, 1)^\top} = \frac{q_{11}x + q_{12}y + q_{13}}{q_{31}x + q_{32}y + q_{33}} \tag{A.16}$$

$$y'_{(x,y)} = \frac{\mathbf{q}_2^\top(x, y, 1)^\top}{\mathbf{q}_3^\top(x, y, 1)^\top} = \frac{q_{21}x + q_{22}y + q_{23}}{q_{31}x + q_{32}y + q_{33}} \tag{A.17}$$

Since a line is still a line under projective transformation, hence

$$\frac{dy'}{dx'} = \frac{y'_{(x+\cos\varphi, y+\sin\varphi)} - y'_{(x-\cos\varphi, y-\sin\varphi)}}{x'_{(x+\cos\varphi, y+\sin\varphi)} - x'_{(x-\cos\varphi, y-\sin\varphi)}} \tag{A.18}$$

$$= \frac{\mathbf{q}_2^\top\mathbf{X}_+\mathbf{q}_3^\top\mathbf{X}_+ - \mathbf{q}_2^\top\mathbf{X}_-\mathbf{q}_3^\top\mathbf{X}_-}{\mathbf{q}_1^\top\mathbf{X}_+\mathbf{q}_3^\top\mathbf{X}_+ - \mathbf{q}_1^\top\mathbf{X}_-\mathbf{q}_3^\top\mathbf{X}_-} \tag{A.19}$$

where

$$\mathbf{X}_+ = (x + \cos\varphi, y + \sin\varphi, 1)^\top \tag{A.20}$$

$$\mathbf{X}_- = (x - \cos\varphi, y - \sin\varphi, 1)^\top \ . \tag{A.21}$$

By equivalent transformations, it can be proved that

$$\mathbf{q}_2^\top\mathbf{X}_+\mathbf{q}_3^\top\mathbf{X}_+ - \mathbf{q}_2^\top\mathbf{X}_-\mathbf{q}_3^\top\mathbf{X}_- = f(\mathbf{q}_2, \mathbf{q}_3, x, y, \varphi) \tag{A.22}$$

$$\mathbf{q}_1^\top\mathbf{X}_+\mathbf{q}_3^\top\mathbf{X}_+ - \mathbf{q}_1^\top\mathbf{X}_-\mathbf{q}_3^\top\mathbf{X}_- = f(\mathbf{q}_1, \mathbf{q}_3, x, y, \varphi) \tag{A.23}$$

where

$$f(\mathbf{u}, \mathbf{v}, x, y, \varphi) = (u_2 v_1 - u_1 v_2)(x \sin\varphi - y \cos\varphi) + (u_1 v_3 - u_3 v_1)\cos\varphi + (u_2 v_3 - u_3 v_2)\sin\varphi \ . \tag{A.24}$$

Therefore,

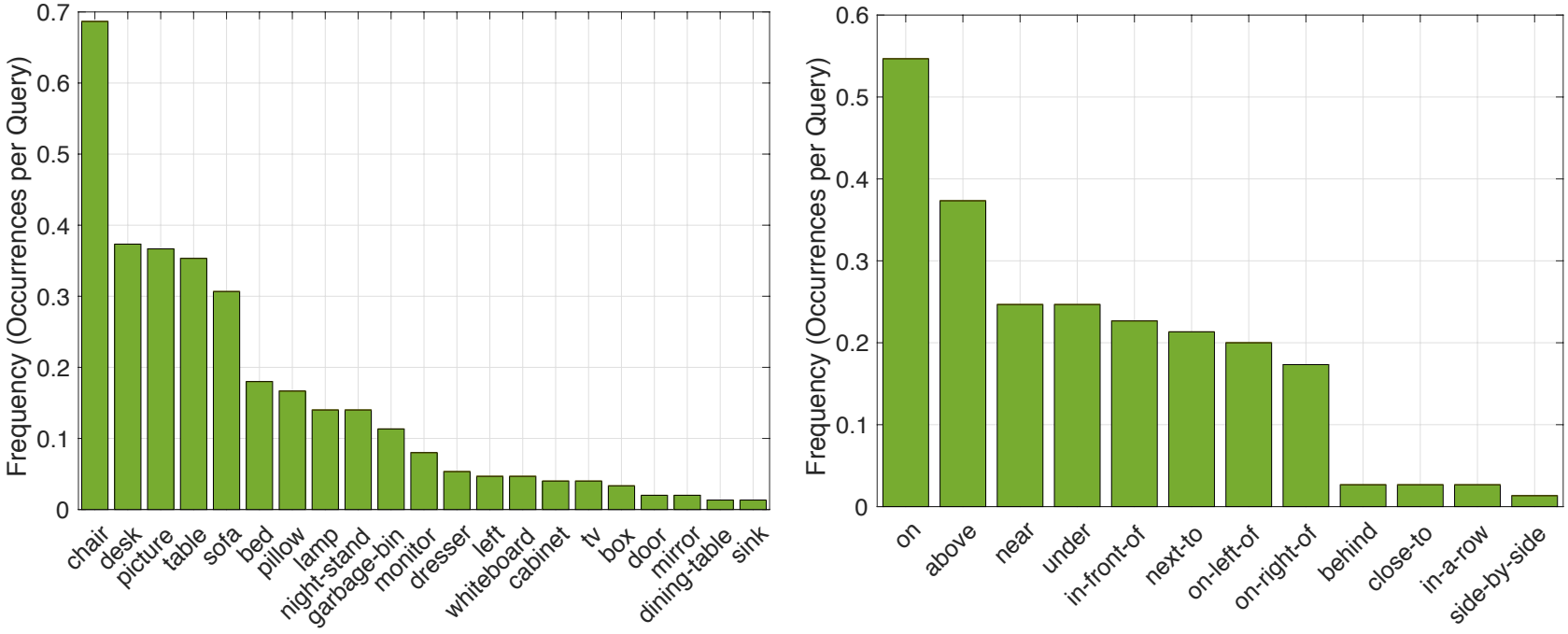
$$\varphi' = \arctan \frac{f(\mathbf{q}_2, \mathbf{q}_3, x, y, \varphi)}{f(\mathbf{q}_1, \mathbf{q}_3, x, y, \varphi)} \ . \tag{A.25}$$

Appendix B: Additional results in abstractive scene representation

B.1 Additional details about datasets

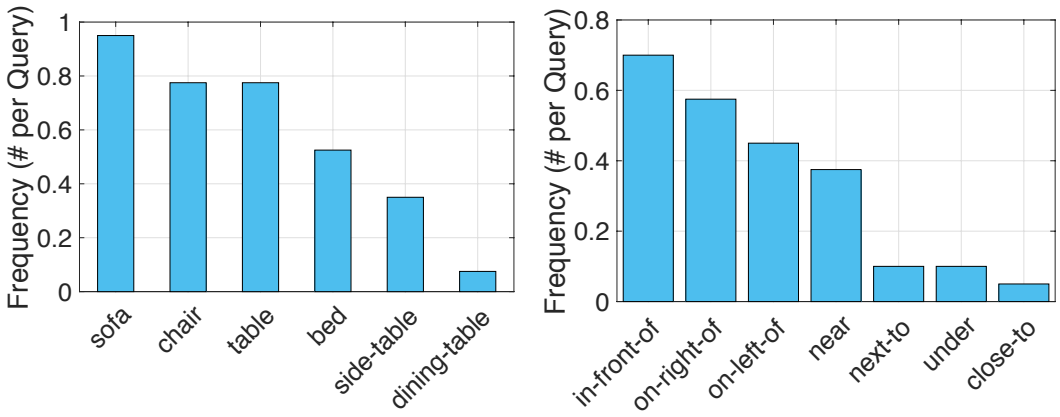
We have 150 of the images annotated in the SUN RGB-D test dataset. We show the statistics about queries used in SUN RGB-D evaluation. In average, SUN RGB-D annotation has 4.26 objects, 2.65 relations, 19.85 words and 2.69 sentences per query. Figure B.1(a) shows the averaged occurrences per query of each object category and Figure B.1(b) shows the averaged occurrences per query of each spatial relation category. The object and relation categories are sorted in the descending order w.r.t. the frequency.

In average, 3DGP annotation has 3.06 objects, 1.94 relations, 17.06 words and 1.94 sentences per query. Similarly, we show the object category and spatial relation frequencies in Figure B.2. Different from the statistics of SUN RGB-D where spatial relation `on` has the highest frequency, spatial relations in 3DGP are mostly horizontal. This is because, for 3DGP, we only have DPM detectors for 6 furniture categories and all of them are on the floor.



(a) (b)

Figure B.1: SUN RGB-D query statistics: (a) frequency (occurrences per query) of objects and (b) frequency (occurrences per query) of spatial relations.



(a) (b)

Figure B.2: Query statistics in 3DGP evaluation: (a) Frequency (occurrences per query) of objects, and (b) frequency (occurrences per query) of spatial relations.

B.2 Additional qualitative retrieval results

We provide additional results in SUN RGB-D for top-3 retrievals in Figure B.3. The ground truth image is shown with a blue bar on its top. Although it happens rare in this evaluation, there are cases when there are images other than the ground truth that meet the descriptions of the query (e.g., the last example in Figure B.3). Qualitative results with matched 3D layout are shown in Figure B.4. The figure shows the 3D layouts with camera location corresponding to the best matched 2D spatial layouts (from 5 layout samples).

Figure B.3: Top 3 retrieved images in SUN RGB-D. Ground truth images appear with **blue** bars on top. Green bounding boxes are detection outputs matching the generated 2D layouts. Red boxes are missing objects (not detected) w.r.t. the expectation of generated 2D layouts.

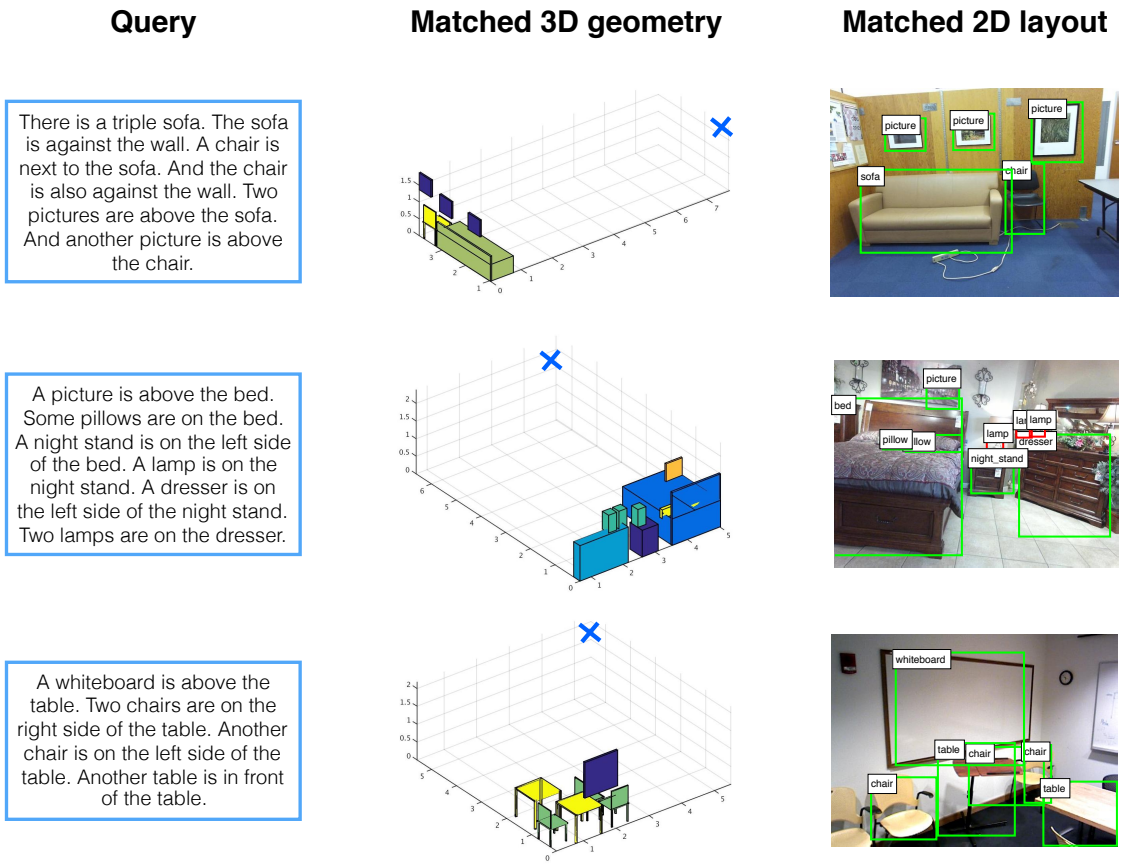
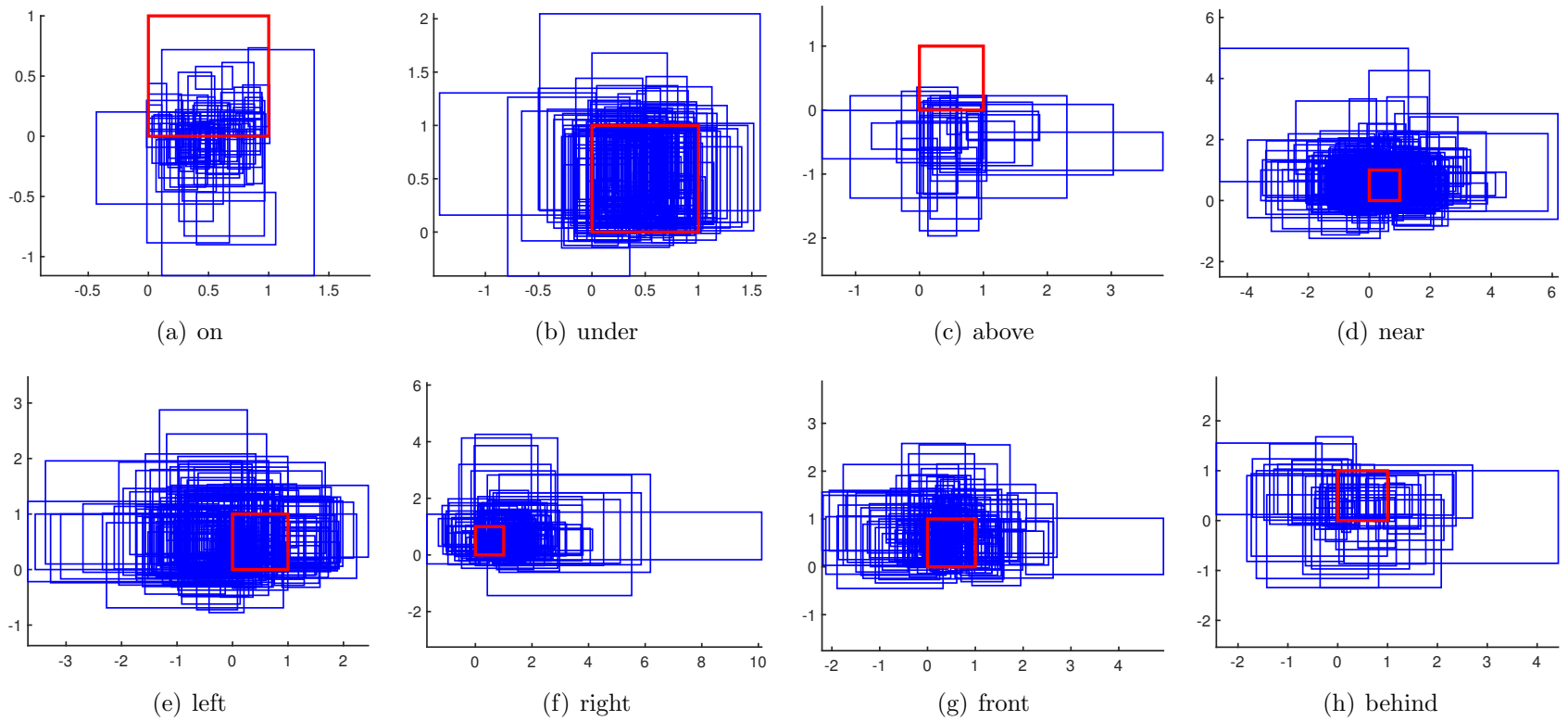


Figure B.4: Matched 3D and 2D layouts based on our greedy 2D layout matching for three ground truth images in SUN RGB-D. Blue crosses represent camera locations. Green bounding boxes are object detection outputs that match the 2D layouts generated from the text queries. Red bounding boxes represent a missing object (not detected by the object detector) within the expected region proposed by 2D layouts.

B.3 Learned 2D spatial relationships in baseline

The learned distributions of 2D spatial relationships in the nearest neighbor baseline algorithm are shown in Figure B.5. The figure shows the relationship between the subject and the object (*subject-relation-object*) w.r.t. all eight atomic spatial relations (other relations are built upon these atomic relations). For each

relationship, the annotated bounding boxes of each pair of subjects and objects are normalized (rescaled in both $x-$ and $y-$ coordinates) so that the subject bounds to a 1×1 square with bottom left $(0, 0)$ and top right $(1, 1)$. All of the normalized relation annotations are visualized in the figure. The nearest neighbor classifier is based on the IOU scores of normalized bounding boxes.



(a) on (b) under (c) above (d) near

(e) left (f) right (g) front (h) behind

Figure B.5: Learned distribution of 2D spatial relations in **subject-relation-object** relationships. Red bounding boxes represent the subject and blue bounding boxes represent the sampled objects in the annotations corresponding to each relation. The subject is normalized to 1×1 squares (with bottom-left $(0, 0)$ and top-right $(1, 1)$) and all objects are rescaled with the same normalization factors in x-y coordinates.

Appendix C: Additional results of visual n-gram models

C.1 Relating images and captions: additional results

As an addition to the image and caption retrieval results on COCO-5K and Flickr-30K presented in the paper, we also provide retrieval results on the COCO-1K dataset, a test set of $1,000$ images provided by Karpathy and Fei-Fei [141]. In Table C.1, we show the caption retrieval (left) and image retrieval (right) performance of four baseline models and our visual n-gram models on COCO-1K. We do not report results we obtained with the last version of the neural image captioning model [185] here because that model was trained on COCO validation set that was used as the basis for the COCO-1K test set.

The results on the COCO-1K dataset are in line with the results presented in the paper: our n-gram model performs roughly on par with recurrent language models [141, 142], but like these language models, it performs worse than models that were developed specifically for retrieval tasks [160, 162].

We provide additional results to demonstrate the effectiveness of end-to-end training. We trained a Jelinek-Mercer model on the ImageNet features as an additional baseline and compare it with the end-to-end Jelinek-Mercer model in COCO-5K. The results are shown in Table C.2 which reveals that an end-to-end trained

Table C.1: Recall@k (for three cut-off levels k) of caption and image retrieval on the COCO-1K dataset for three baseline systems and our visual n-gram models (with and without finetuning). Baselines are separated in models dedicated to retrieval (top) and image-conditioned language models (bottom). Higher is better.

COCO-1K	**Caption retrieval**			**Image retrieval**		
	R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
Retrieval models						
Klein et al. [160]	38.9	68.4	80.1	25.6	60.4	76.8
Wang et al. [162]	50.1	79.7	89.2	39.6	75.2	86.9
Language models						
BRNN [141]	38.4	69.9	80.5	27.4	60.2	74.8
M-RNN [142]	41.0	73.0	83.5	29.0	42.2	77.0
Ours						
Naive n-gram	3.1	9.2	14.6	1.1	4.2	7.3
Jelinek-Mercer	22.5	47.6	60.7	12.8	33.5	46.5
J-M + finetuning	39.9	70.5	82.5	25.4	55.8	70.2

Table C.2: Recall@k (for three cut-off levels k) of caption and image retrieval on the COCO-5K dataset for four variants of our visual n-gram models (with and without finetuning). Higher is better.

COCO-5K	**Caption retrieval**			**Image retrieval**		
	R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
Imagenet + J-M	8.0	21.6	31.2	4.4	14.0	21.5
End-to-end J-M	8.7	23.1	33.3	5.0	14.5	21.9
Imagenet + J-M (finetuning)	12.7	31.0	43.0	6.5	18.9	28.1
End-to-end J-M (finetuning)	17.8	41.9	53.9	11.0	29.0	40.2

Jelinek-Mercer model outperforms the one trained with ImageNet features in both non-finetuning and finetuning modes.

C.2 Phrase prediction: additional results

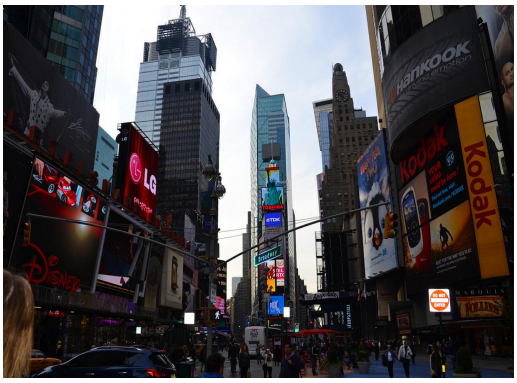
We show additional qualitative results for predicting unigrams and bigrams in Figure C.1 and Figure C.2.

Unigrams	**Bigrams**
Sign	Neon sign
Bar	Motel in
Ave	Store in
Store	Sign for
Diner	Sacramento CA

Ferris	Ferris wheel
Blue	Lafayette Park
Wheel	Coney Island
Lafayette	Blue sky
Tower	Amusement park

Carriage	Horse drawn
Winter	Horse and
Horse	Winter in
Snow	Blizzard of
Blizzard	Snowy day

Figure C.1: Five highest-scoring visual unigrams and bigrams for five images in our test set. From top to bottom, photos are courtesy of: (1) Mike Mozart (CC BY 2.0); (2) owlpacino (CC BY-ND 2.0); and (3) brando.n (CC BY 2.0).



Unigrams	**Bigrams**
Times	Times Square
Shinjuku	Shinjuku Tokyo
Ginza	Manhattan new
Manhattan	Hong Kong
NYC	Eaton Center

Tokyo	Shinjuku Tokyo
Osaka	Tokyo Japan
Shinjuku	Vending machine
Vending	Osaka Japan
Store	Store in

Golden	Golden Gate
Marin	Suspension bridge
Suspension	Mackinac Island
Cruise	Oracle Team
Forth	Brooklyn Bridge

Figure C.2: Five highest-scoring visual unigrams and bigrams for five images in our test set. From top to bottom, photos are courtesy of: (1) Laura (CC BY-NC 2.0); (3) inefekt69 (CC BY-NC-ND 2.0); and (3) Yahui Ming (CC BY-NC-ND 2.0).

Appendix D: License Information for YFCC100M Photos

We reproduce all YFCC100M photos that appear in the main thesis with relevant authorship and license information in Figure D.1, D.2, D.3 and D.4.

Figure D.1: Four high-scoring visual n-grams for three images in our test set according to our visual n-gram model, which was trained *solely* on *unsupervised* web data. We selected the n-grams that are displayed in the figure from the five highest scoring n-grams according to our model, in such a way as to minimize word overlap between the n-grams. From top to bottom, photos are courtesy of: (1) Stuart L. Chambers (CC BY-NC 2.0); (2) Martin Pettitt (CC BY 2.0); (3) Gav Owen (C).

Figure D.2: Four highest-scoring images for n-gram queries "Washington State", "Washington DC", "Washington Nationals", and "Washington Capitals" from a collection of 931,588 YFCC100M test images. Washington Nationals is a Major League Baseball team; Washington Capitals is a National Hockey League hockey team. The figure only shows images from the YFCC100M dataset whose license allows reproduction. From the top-left photo in clockwise direction, the photos are courtesy of: (1) Colleen Lane (CC BY-ND 2.0); (2) Ryaninc (CC BY 2.0); (3) William Warby (CC BY 2.0); (4) Cliff (CC BY 2.0); (5) Boomer-44 (CC BY 2.0); (6) Dannebrog (CC BY-ND 2.0); (7) S. Yume (CC BY 2.0); (8) Bridget Samuels (CC BY-NC-ND 2.0); (9) David G. Steadman (Public Domain Mark 1.0); (10) Hockey Club Torino Bulls (CC BY 2.0); (11) Brent Moore (CC BY-NC 2.0); (12) Andrew Malone (CC BY 2.0); (13) Terren in Virginia (CC BY 2.0); (14) Guru Sno Studios (CC BY-ND 2.0); (15) Derek Hatfield (CC BY 2.0); and (16) Bruno Kussler Marques (CC BY 2.0).

Figure D.3: Four highest-scoring images for n-gram queries "Market Street", "street market", "city park", and "Park City" from a collection of 931,588 YFCC100M images. Market Street is a common street name, for instance, it is one of the main thoroughfares in San Francisco. Park City (Utah) is a popular winter sport destination. The figure only shows images from the YFCC100M dataset whose license allows reproduction. From left to right, photos are courtesy of the following photographers (license details between brackets. **Row 1:** (1) Jonathan Percy (CC BY-NC-SA 2.0); (2) Rachel Clarke (CC BY-NC-ND 2.0); (3) Richard Lazzara (CC BY-NC-ND 2.0); and (4) AboutMyTrip dotCom (CC BY 2.0). **Row 2:** (1) Alex Holyoake (CC BY 2.0); (2) Marnie Vaughan (CC BY-NC 2.0); (3) Hector E. Balcazar (CC BY-NC 2.0); and (4) Marcin Chady (CC BY 2.0). **Row 3:** (1) Rien Honnef (CC BY-NC-ND 2.0); (2) IvoBe (CC BY-NC 2.0); (3) Daniel Hartwig (CC BY 2.0); and (4) Benjamin Chodroff (CC BY-NC-ND 2.0). **Row 4:** (1) Guido Bramante (CC BY 2.0); (2) Alyson Hurt (CC BY-NC 2.0); (3) Xavier Damman (CC BY-NC-ND 2.0); and (4) Cassandra Turner (CC BY-NC 2.0).

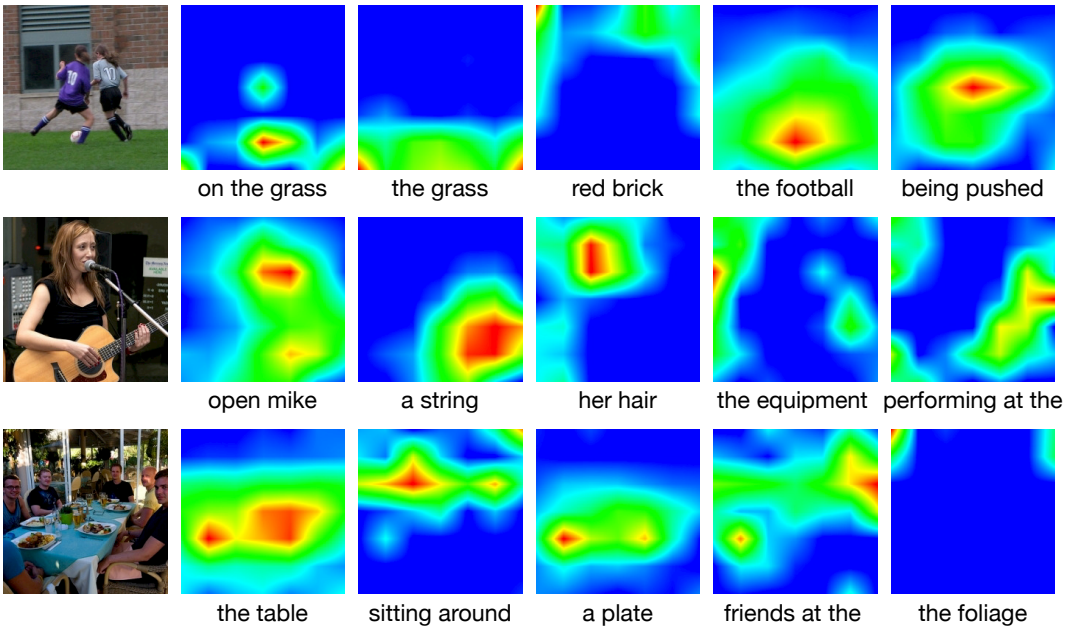


Figure D.4: Discriminative regions of five n-grams for three images, computed using class activation mapping. From top to down, photos are courtesy of the following photographers (license details between brackets. **Row 1:** DebMomOf3 (CC BY-ND 2.0). **Row 2:** fling93 (CC BY-NC-SA 2.0). **Row 3:** Magnus (CC BY-SA 2.0).

Bibliography

[1] Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, and Anton Van Den Hengel. A survey of appearance models in visual object tracking. *ACM Trans. Intell. Syst. Technol.*, 4(4):58:1–58:48, October 2013.

[2] Ang Li, Feng Tang, Yanwen Guo, and Hai Tao. Discriminative nonorthogonal binary subspace tracking. *European Conference on Computer Vision (ECCV)*, pages 258–271, 2010.

[3] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 2, pages 142–149. IEEE Comput. Soc, 2000.

[4] Jianbo Shi and Carlo Tomasi. Good features to track. Technical report, Ithaca, NY, USA, 1993.

[5] Ang Li, Feng Tang, Yanwen Guo, and Hai Tao. Efficient discriminative nonorthogonal binary subspace with its application to visual tracking. *CoRR*, abs/1509.08383, 2015.

[6] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, August 2000.

[7] Z. Yu, A. Li, O. C. Au, and C. Xu. Bag of textons for image segmentation via soft clustering and convex shift. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 781–788, June 2012.

[8] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, August 2004.

[9] Joe Yue-Hei Ng, Matthew J. Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, pages 4694–4702. IEEE Computer Society, 2015.

[10] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[11] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *CoRR*, abs/1609.08675, 2016.

[12] Seymour. Papert. *The summer vision project*. Massachusetts Institute of Technology, Project MAC, [Cambridge, Mass.], 1966.

[13] Annette Herskovits and Thomas O. Binford. On boundary detection. *MIT AI Memo*, 1970.

[14] Berthold K.P. Horn. The binford-horn line-finder. *MIT AI Memo*, 1973.

[15] Early processing of visual information - early processing of visual information. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 275(942):483–519, 1976.

[16] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'77, pages 659–663, San Francisco, CA, USA, 1977. Morgan Kaufmann Publishers Inc.

[17] David Marr and E. Hildreth. Theory of edge detection. *Proc. Roy. Soc. London*, 1980.

[18] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov 1986.

[19] Gregory Randall, Jérémie Jakubowicz, Rafael Grompone von Gioi, and Jean-Michel Morel. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:722–732, 2008.

[20] Chris Harris and Mike Stephens. A combined corner and edge detector. In Christopher J. Taylor, editor, *Alvey Vision Conference*, pages 1–6. Alvey Vision Club, 1988.

[21] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[22] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.

[23] Fei-Fei Li and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '05, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society.

[24] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Efficient learning with sets of features. *J. Mach. Learn. Res.*, 8:725–760, May 2007.

[25] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[26] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, 32(9):1627–1645, Sept 2010.

[27] Saurabh Singh, Abhinav Gupta, and Alexei A. Efros. Unsupervised discovery of mid-level discriminative patches. In *European Conference on Computer Vision*, 2012.

[28] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[29] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015.

[32] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[33] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[34] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sanchez, Patrick Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 34(9):1704–1716, September 2012.

[35] Florent Perronnin and Christopher R. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2007.

[36] Ang Li, Vlad I. Morariu, and Larry S. Davis. Planar structure matching under projective uncertainty for geolocation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[37] Ang Li, Vlad I. Morariu, and Larry S. Davis. Selective encoding for recognizing unreliably localized faces. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[38] Ang Li, Jin Sun, Joe Yue-Hei Ng, Ruichi Yu, Vlad I. Morariu, and Larry S. Davis. Generating holistic 3d scene abstractions for text-based image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[39] Ang Li, Allan Jabri, Armand Joulin, and Laurens van der Maaten. Learning visual n-grams from web data. *CoRR*, abs/1612.09161, 2016.

[40] Ang Li, Allan Jabri, Armand Joulin, and Laurens van der Maaten. Learning Visual N-Grams from Web Data. In *International Conference on Computer Vision (ICCV)*, 2017.

[41] James Hays and Alexei A. Efros. im2gps: estimating geographic information from a single image. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[42] Tsung-Yi Lin, Serge Belongie, and James Hays. Cross-view image geolocalization. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Portland, OR, June 2013.

[43] Georges Baatz, Olivier Saurer, Kevin Köser, and Marc Pollefeys. Large scale visual geo-localization of images in mountainous terrain. In *Proceedings of European Conference on Computer Vision (ECCV)*, volume II, pages 517–530, Florence, Italy, 2012.

[44] B.C. Matei, N. Vander Valk, Zhiwei Zhu, Hui Cheng, and H.S. Sawhney. Image to lidar matching for geotagging in urban environments. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 413–420, Jan 2013.

[45] Ming-Yu Liu, Oncel Tuzel, Ashok Veeraraghavan, and Rama Chellappa. Fast directional chamfer matching. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010.

[46] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. *Theory of Computing*, 8(19):415–428, 2012.

[47] Yunpeng Li, Noah Snavely, and Daniel P. Huttenlocher. Location recognition using prioritized feature matching. In *Proceedings of European Conference on Computer Vision (ECCV)*, volume II, pages 791–804, Heraklion, Crete, Greece, 2010.

[48] D.M. Chen, G. Baatz, K. Koser, S.S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, Xin Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 737–744, Nov 2011.

[49] Grant Schindler, Matthew Brown, and Richard Szeliski. City-scale location recognition. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, 2007.

[50] A.R. Zamir and M. Shah. Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2014.

[51] Yan-Tao Zheng, Ming Zhao, Yang Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, Tat-Seng Chua, and H. Neven. Tour the world: Building a web-scale landmark recognition engine. *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1085–1092, 2009.

[52] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2599–2606, June 2009.

[53] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 15–29. 2012.

[54] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *IEEE Int'l Conf. Computer Vision (ICCV)*, pages 667–674, Nov 2011.

[55] Mayank Bansal, Harpreet S. Sawhney, Hui Cheng, and Kostas Daniilidis. Geo-localization of street views with aerial image databases. In *ACM Int'l Conf. Multimedia (MM)*, pages 1125–1128, 2011.

[56] C. Schmid and A. Zisserman. Automatic line matching across views. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pages 666–, Washington, DC, USA, 1997. IEEE Computer Society.

[57] H. Bay, V. Ferrari, and L. Van Gool. Wide-baseline stereo matching with line segments. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 329–336 vol. 1, June 2005.

[58] Hyunwoo Kim and Sukhan Lee. Wide-baseline image matching based on coplanar line intersections. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1157–1164, Oct 2010.

[59] Lu Wang, U. Neumann, and S. You. Wide-baseline image matching using line signatures. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1311–1318, Sept 2009.

[60] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'77, pages 659–663, San Francisco, CA, USA, 1977. Morgan Kaufmann Publishers Inc.

[61] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 30(7):1270–1281, July 2008.

[62] C.F. Olson. A probabilistic formulation for hausdorff matching. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 150–156, Jun 1998.

[63] A. Elgammal, V. Shet, Y. Yacoob, and L.S. Davis. Exemplar-based tracking and recognition of arm gestures. In *Image and Signal Processing and Analysis, 2003. ISPA 2003. Proceedings of the 3rd International Symposium on*, volume 2, pages 656–661 Vol.2, Sept 2003.

[64] Aswin C. Sankaranarayanan and Rama Chellappa. Optimal multi-view fusion of object locations. In *Proceedings of the 2008 IEEE Workshop on Motion and Video Computing*, WMVC '08, pages 1–8, Washington, DC, USA, 2008. IEEE Computer Society.

[65] R.G. von Gioi, J. Jakubowicz, J. M Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 32(4):722–732, April 2010.

[66] Lacey Best, Hu Han, Charles Otto, Brendan Klare, and Anil K. Jain. Unconstrained face recognition: Identifying a person of interest from a media

collection. Technical Report MSU-CSE-14-1, Department of Computer Science, Michigan State University, East Lansing, Michigan, March 2014.

[67] Dong Chen, Xudong Cao, Fang Wen, and Jian Sun. Blessing of dimisionality: High dimensional feature and its efficient compression for face verification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[68] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A compact and discriminative face track descriptor. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[69] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, ECCV'10, pages 143–156, Berlin, Heidelberg, 2010. Springer-Verlag.

[70] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher Vector Faces in the Wild. In *British Machine Vision Conference (BMVC)*, 2013.

[71] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1891–1898, June 2014.

[72] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[73] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[74] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 529–534. IEEE, 2011.

[75] Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154, May 2004.

[76] M.E. Fathy, V.M. Patel, and R. Chellappa. Face-based active authentication on mobile devices. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 1687–1691, April 2015.

[77] Vidit Jain and Erik Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[78] J. Sanchez, F. Perronnin, T. E. J. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision (IJCV)*, 2013.

[79] F. Perronnin, Yan Liu, J. Sanchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3384–3391, June 2010.

[80] Andrew Wagner, John Wright, Arvind Ganesh, Zihan Zhou, Hossein Mobahi, and Yi Ma. Toward a practical face recognition system: Robust alignment and illumination by sparse representation. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 34(2):372–386, 2012.

[81] John Wright, Allen Y. Yang, Arvind Ganesh, Shankar S. Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 31(2):210–227, 2009.

[82] Thomas Berg and Peter N. Belhumeur. Tom-vs-pete classifiers and identity-preserving alignment for face verification. In *British Machine Vision Conference, BMVC 2012, Surrey, UK, September 3-7, 2012*, pages 1–11, 2012.

[83] Anelia Angelova and Shenghuo Zhu. Efficient object detection and segmentation for fine-grained recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 811–818. IEEE, 2013.

[84] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision (IJCV)*, 2013.

[85] Minh Hoai Nguyen, Lorenzo Torresani, Fernando De la Torre, and Carsten Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2009.

[86] Tian Lan, Yang Wang 0003, and Greg Mori. Discriminative figure-centric models for joint action localization and recognition. In Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc J. Van Gool, editors, *Proceedings of International Conference on Computer Vision (ICCV)*, pages 2003–2010. IEEE, 2011.

[87] Olga Russakovsky, Yuanqing Lin, Kai Yu, and Fei-Fei Li. Object-centric spatial pooling for image classification. In Andrew W. Fitzgibbon, Svetlana

Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Proceedings of European Conference on Computer Vision (ECCV)*, volume 7573 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2012.

[88] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, November 2004.

[89] Justin Johnson, Ranjay Krishna, Michael Stark, Jia Li, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[90] Parisa Kordjamshidi, Martijn Van Otterlo, and Marie-Francine Moens. Spatial role labeling: Towards extraction of spatial relations from natural language. *ACM Trans. Speech Lang. Process.*, 8(3):4:1–4:36, December 2011.

[91] Yong Rui, Thomas S. Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(1):39 – 62, 1999.

[92] Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[93] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D. Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the Fourth Workshop on Vision and Language*, pages 70–80, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[94] C. Lawrence Zitnick, Devi Parikh, and Lucy Vanderwende. Learning the visual interpretation of sentences. In *Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV)*, ICCV '13, pages 1681–1688, Washington, DC, USA, 2013. IEEE Computer Society.

[95] W. Choi, Y. W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[96] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[97] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalanditis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

[98] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[99] Nitish Srivastava and Ruslan Salakhutdinov. Multimodal learning with deep boltzmann machines. *Journal of Machine Learning Research (JMLR)*, 15:2949–2980, 2014.

[100] Behjat Siddiquie, Brandyn White, Abhishek Sharma, and Larry S. Davis. Multi-modal image retrieval for complex queries using small codes. In *International Conference on Multimedia Retrieval (ICMR)*, 2014.

[101] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 2008.

[102] Y. Xiang and S. Savarese. Object detection by 3d aspectlets and occlusion reasoning. In *International Conference on Computer Vision Workshops (ICCVW)*, Dec 2013.

[103] M. Zeeshan Zia, Michael Stark, and Konrad Schindler. Towards scene understanding with detailed 3d object representations. *International Journal of Computer Vision*, 112(2):188–203, 2015.

[104] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *International Conference on Computer Vision (ICCV)*, pages 2650–2658, Dec 2015.

[105] Derek Hoiem and Silvio Savarese. *Representations and Techniques for 3D Object Recognition and Scene Interpretation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.

[106] Luca Del Pero, Joshua Bowdish, Daniel Fried, Bonnie Kermgard, Emily Hartley, and Kobus Barnard. Bayesian geometric modeling of indoor scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2719–2726. IEEE Computer Society, 2012.

[107] L. Del Pero, J. Bowdish, B. Kermgard, E. Hartley, and K. Barnard. Understanding bayesian rooms using composite 3d object models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 153–160, June 2013.

[108] Bob Coyne and Richard Sproat. Wordseye: An automatic text-to-scene conversion system. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 487–496, New York, NY, USA, 2001. ACM.

[109] Angel Chang, Manolis Savva, and Christopher D. Manning. Learning spatial knowledge for text to 3d scene generation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2028–2038, Doha, Qatar, October 2014. Association for Computational Linguistics.

[110] T Kulkarni, Ilker Yildirim, Pushmeet Kohli, W Freiwald, and Joshua B Tenenbaum. Deep generative vision as approximate bayesian computation. In *NIPS 2014 ABC Workshop*, 2014.

[111] Tejas D. Kulkarni, Vikash K. Mansinghka, Pushmeet Kohli, and Joshua B. Tenenbaum. Inverse graphics with probabilistic CAD models. *CoRR*, abs/1407.1339, 2014.

[112] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 127–135. Curran Associates, Inc., 2015.

[113] John M. Snyder. Interval analysis for computer graphics. *SIGGRAPH Comput. Graph.*, 26(2):121–130, July 1992.

[114] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

[115] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[116] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Proceedings of European Conference on Computer Vision*, 2016.

[117] Ross Girshick. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision (Proceedings of International Conference on Computer Vision (ICCV))*, 2015.

[118] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 2006.

[119] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 2015.

[120] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and L. Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[121] A. Bendale and T. Boult. Towards open world recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[122] T.L. Berg and D.A. Forsyth. Animals on the web. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[123] X. Chen and A. Gupta. Webly supervised learning of convolutional networks. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[124] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from internet image searches. *Proceedings of the IEEE*, 2010.

[125] A. Joulin, L.J.P. van der Maaten, A. Jabri, and N. Vasilache. Learning visual features from large weakly supervised data. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[126] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[127] S. Vijayanarasimhan and K. Grauman. Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[128] X.-J. Wang, L. Zhang, X. Li, and W.-Y. Ma. Annotating images by mining image search results. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 2008.

[129] B. Thomee, D.A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[130] F. Jelinek and R.L. Mercer. Interpolated estimation of markov source parameters from sparse data. In *Workshop on Pattern Recognition in Practice*, 1980.

[131] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *ICASSP*, 1995.

[132] Kevin Tang, Armand Joulin, Li-Jia Li, and Li Fei-Fei. Co-localization in real-world images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[133] L.-J. Li and L. Fei-Fei. Optimol: Automatic online picture collection via incremental model learning. *International Journal of Computer Vision (IJCV)*, 2010.

[134] Q. Li, J. Wu, and Z. Tu. Harvesting mid-level visual concepts from large-scale internet images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[135] E. Denton, J. Weston, M. Paluri, L. Bourdev, and R. Fergus. User conditional hashtag prediction for images. In *KDD*, 2015.

[136] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[137] H. Izadinia, B.C. Russell, A. Farhadi, M.D. Hoffman, and A. Hertzmann. Deep classifiers from image tags in the wild. In *Proceedings of the 2015 Workshop on Community-Organized Multimodal Mining: Opportunities for Novel Solutions*, pages 13–18. ACM, 2015.

[138] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2010.

[139] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 22(12):1349–1380, 2000.

[140] J. Donahue, L.A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[141] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[142] J. Mao, W. Xu, Y. Yang, J. Wang, and A.L. Yuille. Deep captioning with multimodal recurrent neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[143] H. Fang, S. Gupta, F.N. Iandola, R. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J.C. Platt, C.L. Zitnick, and G. Zweig. From captions to visual concepts and back. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[144] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[145] R. Lebret, P.O. Pinheiro, and R. Collobert. Phrase-based image captioning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

[146] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[147] A. Farhadi, M. Hejrati, M.A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2010.

[148] A. Karpathy, A. Joulin, and L. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[149] S. Li, G. Kulkarni, T.L. Berg, A.C. Berg, and Y. Choi. Composing simple image descriptions using web-scale n-grams. In *CoNLL*, 2011.

[150] A. Passos, V. Kumar, and A. McCallum. Lexicon infused phrase embeddings for named entity resolution. In *CoNLL*, 2014.

[151] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *arXiv:1607.01759*, 2016.

[152] D. Tang, F. Wei, B. Qin, M. Zhou, and T. Liu. Building large-scale twitter-specific sentiment lexicon: A representation learning approach. In *COLING*, 2014.

[153] S. Wang and C. D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2012.

[154] J. Zhang, S. Liu, M. Li, M. Zhou, and C. Zong. Bilingually-constrained phrase embeddings for machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.

[155] W.Y. Zou, R. Socher, D.M. Cer, and C.D. Manning. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[156] S.F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 1996.

[157] J.T. Goodman. A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403–434, 2001.

[158] T. Brants, A.C. Popat, P. Xu, F.J. Och, and J. Dean. Large language models in machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 858–867, 2007.

[159] Y. Bengio and J.-S. Senecal. Quick training of probabilistic neural nets by importance sampling. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2003.

[160] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Fisher vectors derived from hybrid gaussian-laplacian mixture models for image annotation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[161] F. Yan and K. Mikolajczyk. Deep correlation for matching images and text. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[162] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[163] R. Socher, A. Karpathy, Q. Le, C. D. Manning, and A. Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2014.

[164] J. R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539, 2014.

[165] P. Anderson, B. Fernando, M. Johnson, and S. Gould. SPICE: Semantic propositional image caption evaluation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[166] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.

[167] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[168] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[169] Z. Zhang and V. Saligrama. Zero-shot recognition via structured prediction. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[170] A. Farhadi, I. Endres, D. Hoiem, and D.A. Forsyth. Describing objects by their attributes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[171] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[172] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *EACL 2014 Workshop on Statistical Machine Translation*, 2014.

[173] T. Mikolov. *Statistical Language Models based on Neural Networks.* PhD thesis, Brno University of Technology, 2012.

[174] R.R. Selvaraju, A. Das, R. Vedantam, and M. Cogswell. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. In *arXiv Preprint 1610.02391*, 2016.

[175] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[176] A. Jabri, A. Joulin, and L.J.P. van der Maaten. Revisiting visual question answering baselines. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[177] B. Sturm. A simple method to determine if a music information retrieval system is a horse. *IEEE Transactions on Multimedia*, 16(6):1636–1644, 2014.

[178] B. Sturm. Horse taxonomy and taxidermy. HORSE2016, 2016.

[179] A. Torralba and A.A. Efros. Unbiased look at dataset bias. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1528, 2011.

[180] Slava M Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *ICASSP*, 1987.

[181] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C.L. Zitnick, and D. Parikh. VQA: Visual question answering. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[182] L. Yu, E. Park, A.C. Berg, and T.L. Berg. Visual madlibs: Fill in the blank description generation and question answering. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[183] D. Geman, S. Geman, N. Hallonquist, and L. Younes. Visual Turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112(12):3618–3623, 2015.

[184] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L. Jia-Li, D. Ayman Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision (IJCV)*, 2016.

[185] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):652–663, April 2017.