SOCIAL AND EGOCENTRIC IMAGE CLASSIFICATION FOR SCIENTIFIC AND PRIVACY APPLICATIONS

Mohammed Korayem

Submitted to the faculty of the University Graduate School

in partial fulfillment of the requirements

for the degree

Doctor of Philosophy

in the School of Informatics and Computing,

Indiana University

July 2015

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Doctoral Committee

Assistant Professor David J. Crandall, PhD, Committee Chair

Assistant Professor Apu Kapadia,PhD

Associate Professor Johan Bollen,PhD

Associate Professor Predrag Radivojac,PhD

May 26, 2015

In memory to my late father, God bless his soul. Who has always stood behind me and believed in me like none other.

To my mother, I could not have done this without your support and prayers.

To my beloved wife, Maha, whose sacrificial care for me and our children made it possible for me to complete this work and to my children, Yusef, Adam, and Sarah.

Last but not least to my brothers and sister whom I love dearly.

ACKNOWLEDGMENTS

I would like to thank my advisor David Crandall for all his support and what he taught me during my study in Indiana University. I am extremely grateful for time, and effort you spent for me. You are one of the dearest friends to me. I am extremely lucky to have David Crandall as my advisor. Many Thanks to Apu Kapdia, Robert Templeman, Haipeng Zhang, Jingya Wang, and Muhammad Abdul-Mageed for amazing collaboration and nice papers we published together during the PhD program.

I would like to thank Stefan Lee and Sven Bambach for their helpful feedback and comments to improve this work. Thanks for Amr Sabry and Predrag Radivojac for their helping and advising during their leading of the computer science graduate program.

Thanks for my research committee members for their comments and helping.

Thanks to Khalifeh Al Jadda, Trey Grainger and Camilo Ortiz for the nice work we did during my intern in CareerBuilder.com.

This work was supported in part by the National Science Foundation through grants IIS-1253549 and CNS-1408730, Google, and the IU Vice President for Research through its IUCRG and FRSP programs. It used compute facilities provided by NVidia, the Lilly Endowment through its support of the IU Pervasive Technology Institute, the Indiana METACyt Initiative, and NSF (CNS-0521433).

Mohammed Korayem

SOCIAL AND EGOCENTRIC IMAGE CLASSIFICATION FOR SCIENTIFIC AND PRIVACY APPLICATIONS

Image classification is a fundamental computer vision problem with decades of related work. It is a complex task and is a crucial part of many applications. The vision community has created many standard data sets for object recognition and image classification. While these benchmarks are created with the goal of being a realistic, representative sample of the visual world, they often contain implicit biases relating to how the images were selected (as well as which were ignored).

In this thesis, we present two lines of work that apply and test image classification in much more realistic problems. We present systems that utilize image classification, deep learning and probabilistic models on large-scale, realistic, unconstrained, and automatically collected datasets. These capture a wider breadth of life on Earth than conventional datasets due to their scale and diversity. Besides these new datasets and the image classification systems developed, the novel applications we present are interesting in their own right.

The first line of work explores the potential of social media imagery to power large-scale scientific analysis. We focus on two prototype problems motivated by ecology: automatically detecting snowfall and vegetation. Using over 200 million Flickr images, each representing a rich description of the world at a specific time and place. Our results indicate that a combination of text mining techniques and image classification can produce high quality data for scientists from large-scale, noisy social images.

The second line of work addresses privacy concerns related to wearable cameras, by automatically detecting private imagery. We present two systems that focus on differ-

ent aspects of what makes an image private. The first is PlaceAvoider, which recognizes images taken in sensitive places such as bedrooms or bathrooms. The second is ObjectAvoider, which tries to detect key objects that may signal private content.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 COMPUTER VISION AND IMAGE CLASSIFICATION

The goal of computer vision is to automatically infer semantic information from images, by introducing computational methods and techniques. Computer vision has many potential applications across different domains [19,33,54,74,99,110,115,127]. For example, computer vision can be used in industrial robots and parts inspections for quality assurance [127]. In retail applications, computer vision systems could automatically detect products in shopping carts for faster check out [74]. Medical images could be used to create visual representation of the inside of a body for clinical analysis [99]. Computer vision has useful applications in security, including biometric identification and surveillances [110,115].

While some of these applications have become reality, most currently work only in constrained environments and under specific conditions, such as those in Figure 1.1. For instance, the most widely successful application in retail simply detects if there are items in the lower part of the cart, which is a constrained environment where the view is very specific and objects are in front of the camera. To realize the true potential of computer vision, we need new systems that can work well in unconstrained environments on challenging vision problems.

A specific computer vision problem which is a basic building block of many applica-

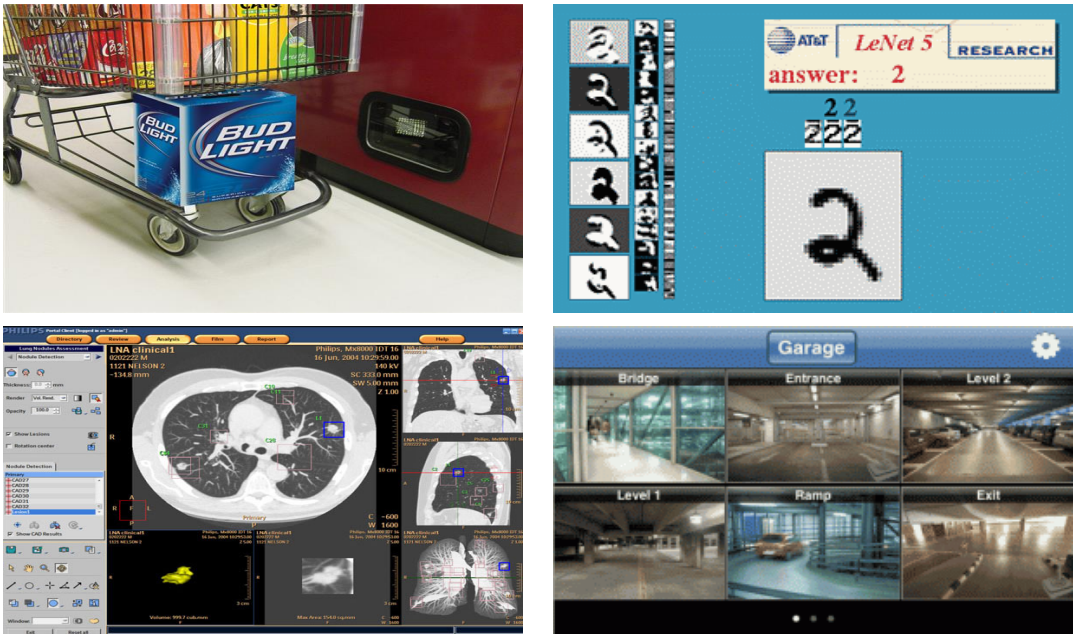


Figure 1.1: Some applications of computer vision. Top left: Retail [6]; top right: hand written recognition [4]; bottom left: medical images [5], and bottom right: surveillance and security application [7].

tions is image classification. The goal of *image classification* is to assign one or more predefined labels to an image based on its visual content. The interpretation and number of labels are application-dependent. This number varies from very small (e.g., 2) in some applications to hundreds in others. The labels can represent both concrete visual concepts, such as presence or absence of objects, as well as high-level semantic concepts, such as popularity or sentiment. One traditional and important application of image classification is image retrieval, where the goal is to retrieve images with specific visual content from image data sets [117, 136].

Image classification has decades of related work [21, 72, 82, 92, 103, 104, 137]. The computer vision community has developed discriminative visual features as well as sophisticated methods for image classification. In order to evaluate the relative power of these techniques, the community has created many standard datasets for testing object recog-

Figure 1.2: Examples from Caltech101 dataset for different objects.

nition and image classification. But while these benchmarks are created with the goal of providing datasets that reflect realistic computer vision applications, they often contain implicit biases relating to how the images were selected (as well as which were ignored) [132]. The Caltech101 dataset [45] stands out as a prominent example, with most images centered and cropped around the object of interest as shown in Figure 1.2. It is difficult to argue that the performance on these constrained datasets reflects the capabilities of algorithms on real world applications of image classification; e.g., real photos taken by consumers.

This means that wherever a new vision technique is developed, it may beat existing techniques on these standard datasets, but it is unclear whether it is a real step forward or is simply better at learning biases of the dataset. For example, recently deep learning (i.e. Convolutional Neural Networks (CNNs)) [82] has gained a lot of attention in the vision community due to its impressive performance on vision problems including image classification. For instance, it outperformed all other techniques in the ImageNet challenge, probably the most famous object category detection competition but one that uses simple

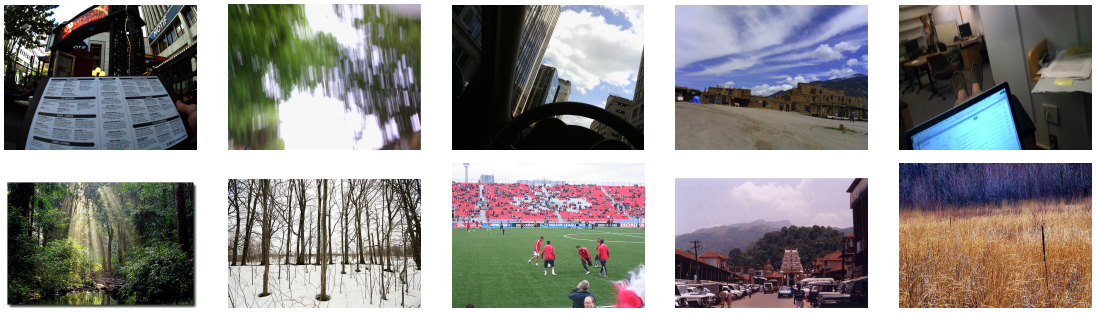


Figure 1.3: Examples from our dataset. Top: life-logging images, bottom: social images from our collected Flickr data sets.

images such as those in Figure 1.2 [112]. Thus, despite deep learning's strong performance on standard data sets, it is still not clear how well it will perform in real-world applications. For example, the very large number of parameters in a CNN introduces the risk of overfitting to dataset bias. Evaluating these techniques on large scale, realistic data sets will be an important step to assess their performance.

In this thesis, we present two lines of work that apply image classification and deep learning to varied problem areas. We present frameworks utilizing image classification and deep learning to solve challenging problems on novel large-scale, unconstrained, and automatically collected datasets. These approaches target new problems in fields outside the ones that computer vision has traditionally studied. The first attempts to crowd-source scientific data collection by automatically collecting and annotating natural phenomenon in public social media imagery. The second helps to ease increasingly important privacy concerns related to life-logging devices and wearable cameras such as Autographer and the Narrative Clip by automatically detecting private imagery. The datasets that arise from these projects capture a wider breadth of life on Earth than conventional datasets due to the sheer scale of the datasets as well as the diversity of imaging sources. These images, such as those in Figure 1.3, reflect the real scenes that people encounter through

their daily lives, in contrast with the simple, clean and unrealistic standard datasets such as those in Figure 1.2. Besides providing novel, realistic, and large-scale datasets for testing classification algorithms, the applications we present are interesting in their own right.

In the first line of work, we explore the potential of mining social media imagery to study ecology phenomena, in particular estimating snowfall and vegetation. The dataset for these tasks aggregates over 200 million images from Flickr, each representing a rich description of the world at a specific time and place. These data points consist of image data, time stamps, global position, and user provided text tags. Our results indicate that a combination of textual data mining techniques and image classification can succeed in producing high quality data sources for scientists from large-scale, noisy input data.

In the second line of work, we consider the implications of a new style of photography using wearable cameras. With the rise in popularity of wearable camera devices like the Narrative Clip, Autographer, and Samsung's Galaxy Gear Smartwatch, concerns over personal privacy have also increased. These devices have the capacity to capture and store more of our private lives than ever before and manually sorting through this increasingly vast data to remove private moments could become nearly impossible. We seek to automate this process by identifying private imagery from life-logging streams using systems that utilize image classification and a probabilistic model to take advantage of the temporal consistencies in the data streams. We present two systems based on this work, each of which focuses on a different aspect of what makes an image private. The first is PlaceAvoider, which tries to recognize images taken in sensitive places such as bedrooms or bathrooms. The second is ObjectAvoider, which looks for certain objects that indicate whether an image is to be kept private; for example, the wearable camera of a user working on screen is likely to capture private data. Our results show that image classification sys-

tems show potential to provide users with automatic, fine-grained privacy controls. The following sections give an overview of these lines of work.

1.2 MINING PHOTO-SHARING SOCIAL MEDIA TO STUDY ECOLOGY PHENOMENA

In recent years, the popularity of social networking websites has increased dramatically. Photo-sharing sites have become particularly popular: Flickr and Facebook alone have collected more than 500 billion images, with over 300 million new images uploaded every day [81, 105]. Millions of people use these sites to share their photos with family and friends, but in the process they are creating huge collections of public online data that contain information about the world. Each photo can be seen as a visual observation of the world at a particular point in time and space. For instance, many (if not most) outdoor images contain some information about the state of the natural world, such as the weather conditions and the presence or absence of plants and animals (Figure 1.4). The aggregation of these millions of photos is observing and capturing the visual world across time and space.

These billions of photos on these sites combined with metadata including timestamps, geo-tags, and captions are a rich unexploited source of information about the state of the world (especially the natural world) and how it is changing over time [34]. These photos could be analyzed to create a new source of data for biologists and ecologists.

Where are marigolds blooming today, and how is this geospatial distribution different from a year ago? Are honeybees less populous this year than last year? Which day do leaves reach their peak color in each county of the northeastern U.S.?

These questions can be addressed to some extent by traditional data collection tech-

Figure 1.4: Many Flickr images contain evidence about the state of the natural world, including that there is snow on the ground at a particular place and time, that a particular species of bird or animal is present, and that particular species of plants are flowering.

niques like satellite instruments, aerial surveys, or longitudinal manual surveys of small patches of land, but none of these techniques allows scientists to collect fine-grained data at continental scales: satellites can monitor huge areas of land but cannot detect fine-grained features like blooming flowers, while manual surveys can collect high-quality and fine-grained data only in a small plot of land. Large-scale analysis of photos on social media sites could provide an entirely new source of data at a fraction of the cost of launching a satellite or hiring teams of biologist observers.

The scientific community, particularly scientists who study climate change, is in need of real-time, global-scale information on the state of the world. Recent work shows that

global climate change is impacting a variety of flora and fauna at local, regional and continental scales: as an example, species of high-elevation and cold-weather mammals have moved northward, some species of butterflies have become extinct, waterfowl are losing coastal wetland habitats as oceans rise, and certain fish populations are rapidly declining [109]. Monitoring these changes is very difficult because it is intractable to collect detailed biological data at global scales. Biologists performed plot-based studies through observing how small patches of land change over time, but this gives information only for a very local area. Meanwhile satellites and aerial surveillance can be used to collect data over large land areas. That can be done for some ecological information like weather patterns, but not for tracking other information like species presence or migration patterns. Also, aerial data has major limitations (e.g., cloud cover, heavy forest cover, and atmospheric conditions and mountain shadows can interfere with the observations).

There are two main challenges to recognize the ecological information latent in these photo datasets. The first is how to recognize ecological phenomena appearing in photos and how to map these observations to specific places and times. Fortunately, modern photo-sharing sites like Flickr and Facebook collect a rich information about photos, including metadata recorded by the digital camera (exposure settings and timestamps) as well as information generated during social sharing (e.g, text tags, comments, and ratings). Recently, photo sharing sites have introduced geo-tag features which record the latitude-longitude coordinates of where on Earth a photo was taken. These geo-tag features are produced either by a GPS receiver on the camera or smartphone, or input manually by the user. Thus online photos include the necessary information to produce geo-temporal data about the world, including information about content (tags and comments), and when (timestamp) and where (geotag) each photo was taken.

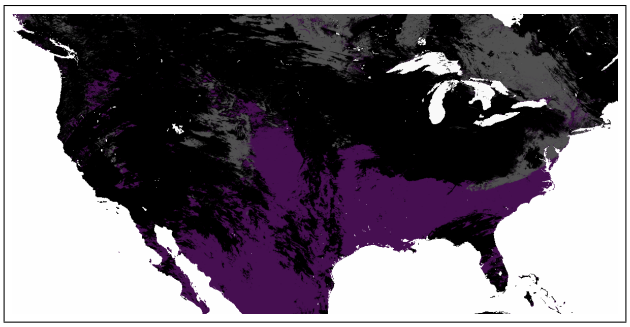
The second key challenge is how to deal with the many sources of biases and noise that exist in online data. People do not photograph all areas of the earth with the same frequency, so there are disproportionate concentrations of activity in cities and tourist attractions. Moreover, photo metadata is often noisy or inaccurate. For instance users sometimes carelessly tag photos. Even if photos are technically tagged correctly, the tags or even visual content of images may be misleading: the tag "snow" on an image may refer to a snow lily or a snowy owl, while snow appearing in an image might be of an indoor zoo and not naturally-occurring.

In this work we present a system to mine data from photo-sharing websites to produce crowd-sourced observations of ecological phenomena. This work can be seen as a first step towards the longer-term goal of mining for many types of phenomena. We study two types of phenomena: ground snow cover and vegetation cover ("green-up") data. Of course, snow and vegetation cover can already be monitored through satellites and weather stations (although neither of these sources is perfect: weather stations are sparse in rural areas and satellites typically cannot observe snow cover through clouds [55, 84]), so this is not a transformative application in and of itself. Instead, these applications are interesting precisely because fine-grained ground truth is available, so that we can evaluate the performance of our crowd-sourced data mining techniques at a very large scale, including thousands of days of data across an entire continent.

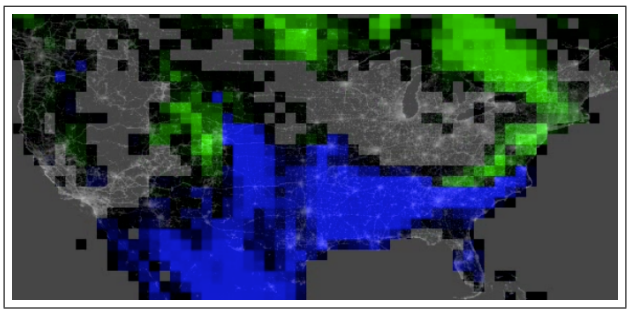
More generally, this work can be seen as a step towards answering a more basic question: How reliable could passive mining of social sharing sites be in producing observations of the world? Analyzing data from social networking and microblogging websites to make estimations and predictions about world events has become a popular research direction, including for example tracking the spread of disease [50], monitoring for fires

and other emergencies [39], predicting product adoption and election outcomes [70], and inferring aggregate public mood [18,106]. In most of these studies, however, there is either no ground truth to judge the quality of the estimates, or the ground truth that is used is an indirect proxy (e.g. since no aggregate public mood data exists, Connor *et al.* oconnor10mood evalute against opinion polls, while Bollen *et al.* [18] compares to stock market indices). In contrast, for predicting some ecological phenomena like vegetation and snow cover, we have daily, dense ground-truth data for the entire globe in the form of satellite observations.

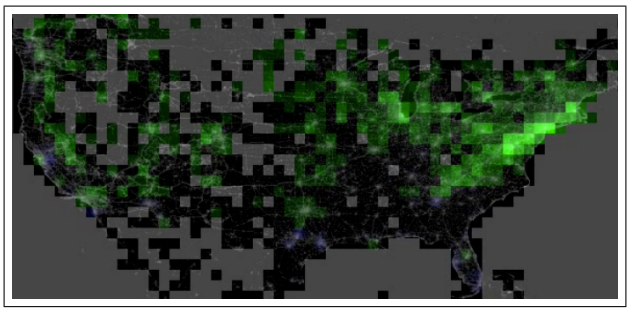
Our system has two components. The first is to utilize image classification and deep learning to deal with the first challenge to recognize ecological phenomena appearing in photos. We initially expected detecting snow in images was an easy problem, in which just looking for large white regions would work reasonably well. However, amongst the hundreds of papers on object and scene classification in the literature, we were surprised to find very few that have explicitly considered detecting snow. A few papers on scene classification include snow-related categories [91, 92, 142], while a few older papers on natural material detection [21,98] consider it along with other categories. We could not find any suitable data set for snow detection. To tackle this problem, we created a new realistic dataset of several thousand images from Flickr with labeled ground truth and applied a variety of recognition techniques including deep learning, we hope our dataset will help spark interest in this somewhat overlooked vision problem. The second component in our framework is a probabilistic model to deal with the noisy data. We apply our methods on a dataset of nearly 200 million geo-tagged Flickr photos to study whether this data can potentially be a reliable resource for scientific research. An example comparing ground truth snow cover data with the estimates produced by our Flickr analysis on one particular



Raw satellite map



Map estimated by Flickr photo analysis



Coarsened satellite map

Figure 1.5: Comparing MODIS satellite snow coverage data for North America on Dec 21, 2009 with estimates produced by analyzing Flickr tags (best viewed in color). *Top:* Original MODIS snow data, where white corresponds with water, black is missing data because of cloud cover, grey indicates snow cover, and purple indicates no significant snow cover. *Middle:* Satellite data coarsened into 1 degree bins, where green indicates snow cover, blue indicates no snow, and grey indicates missing data. *Bottom:* Estimates produced by the Flickr photo analysis proposed in this paper, where green indicates high probability of snow cover, and grey and black indicate low-confidence areas (with few photos or ambiguous evidence).

day (December 21, 2009) is shown in Figure 1.5. Note that the Flickr analysis is sparse in places with few photographs, while the satellite data is missing in areas with cloud cover, but they agree well in areas where both observations are present. This (and the much more extensive experimental results presented later in the thesis) suggests that Flickr analysis may produce useful observations either on its own or as a complement to other observational sources.

To summarize, the main contributions of this line of work include:

— introducing the novel idea of mining photo-sharing sites for geo-temporal information about ecological phenomena

— introducing image classification framework for deriving crowd-sourced observations from noisy, biased data using textual and visual information and

— evaluating the ability of our framework to accurately measure these phenomena, using dense large-scale ground truth.

1.3 MAINTAINING PRIVACY OF FIRST PERSON CAMERA USERS

Digital cameras and camera-enabled devices are now ubiquitous. Cameras are everywhere, including laptops, tablets, smartphones, monitors, gaming systems, televisions, etc. Recently, wearable cameras like iRON snap-cam [67], the Narrative Clip [100], and Autographer [13] have started to become popular. Figure 1.6 shows samples from these devices and Figure 1.7 shows a sample of images taken by one of these life-logging cameras.

These wearable devices enable applications to take photos and other sensor data continuously. For instance, Narrative Clip takes a picture every 30 seconds. These applications allow people to record their lives and capture moments that could not be captured other-

Figure 1.6: Wearable camera devices. Left: Narrative Clip takes photos every 30 seconds; Middle: iON snap camera captures images as well as videos. Right: Autographer has a wide-angle camera and various sensors; (Photos by Narrative, Gizmodo, and Gizmag.)

wise. Also, these devices have many useful applications in safety and health like treating memory loss [29, 63] and crowd-sourced health care [73].

Wearable devices bring innovative applications but also come with many privacy and legal risks [10]. These cameras capture personal and sensitive information of their users as well as people around them [64, 65]. In collecting thousands of images per day and in applications that automatically upload images to the cloud (many wearable devices provide this feature), it would not be feasible to ask users to manually review these images to remove the private ones. So we need to build algorithms and techniques to detect sensitive images and take suitable actions to maintain users' privacy.

Detecting sensitive images is a very hard problem because it includes detecting and reasoning about image content, user activity, environmental context, social norms, etc. However, Hoyle et al.'s study of lifelogging users [64, 65] found that location and the presence of specific objects (especially computer monitors), are main concerns for people's privacy in lifelogging images.

We present two different systems to handle these two problems for first-person camera users. Our work can be seen as an initial step towards building computer vision systems

Figure 1.7: A sampling of images from our lifelogging streams dataset. Three of the nine images include computer or cell phone displays, which often contain potentially sensitive information.

that can be combined with (minimal) human interaction to identify potentially sensitive images. We present two specific systems:

- PlaceAvoider [128] analyzes images to determine where they were taken, and to filter out images from places such as bedrooms and bathrooms, and
- ObjectAvoider [80] filters images based on their content, looking for objects that may signal privacy violations (e.g., computer monitors).

1.3.1 PLACEAVOIDER

The goal of first system, PlaceAvoider, is to detect sensitive places. This framework has two components: image classification and a probabilistic component [128]. For the im-

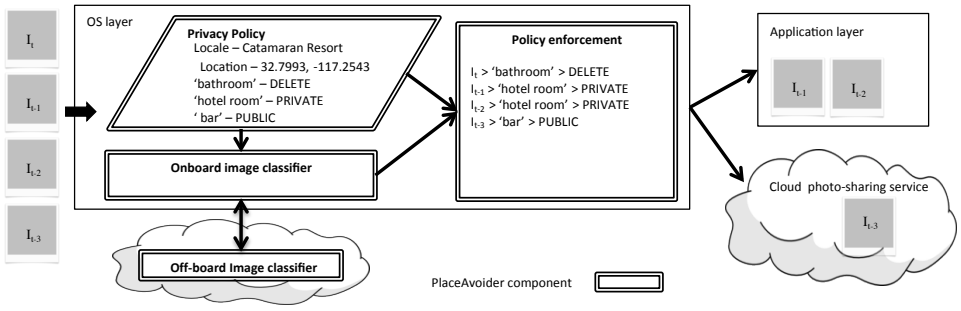


Figure 1.8: A system architecture of PlaceAvoider to enforce a fine-grained camera privacy policy. Our model leverages cloud computation to perform compute-intensive tasks. Cloud-based implementations of PlaceAvoider could also enforce privacy preferences for photo sharing sites [128].

age classification components we use three different types of classifiers: one is based on fine-grained image features (which we call the local classifier) and the second builds visual models for places using different combinations of coarse-grained, scene-level features. The third and last classifier is deep learning which uses Convolutional Neural Networks (CNN) [108] to extract features and classify the images. To our knowledge, we are the first to study this problem (indoor place recognition of life-logging images) and ours is the first data set collected for indoor place recognition in first-person images.

In the second component of our framework, we present a probabilistic classifier that benefits from the weak temporal information associated with life-logging images, and specifically the sequential nature of life-logging images. We show how this streaming classifier is effective and boosts the performance of the classification results. Figure 1.8 show the architecture of our system. The system recognizes images taken in these private spaces to flag them for review before they are made available for applications or sharing on social networks.

1.3.2 OBJECTAVOIDER

PlaceAvoider is designed to detect sensitive images based on where they were taken. However, many images might be taken in public 'non-private' spaces, but still may contain sensitive objects, in particular computer screens [64, 65]. Our ObjectAvoider system is designed to handle this issue. We start by recognizing computer screens in first-person images, and then we try to recognize sensitive applications running on the monitors. Hence, we have two different classifiers: a screen classifier that detects the presence of computer screen in images, while an application classifier that tries to recognize the applications "displayed" on the screen.

1.4 SUMMARY OF THESIS AND CONTRIBUTIONS

The main objective for our work is to build image classification systems for large-scale, unconstrained, and automatically collected datasets. We present two lines of work. In the first one, we explore the potential of mining social media imagery to study ecology phenomena. The goal of the second line of our work is to detect private imagery of first person camera users. To achieve these objectives we pose the answer to the following research questions:

- What types of visual features work best for natural scene classification in realistic, large scale social image collections, in particular for detecting snow and vegetation, and how does their performance compare to user-generated text tags?
- How can the implicit noise in social images be modeled and mediated?
- How well do deep learning techniques perform on large-scale, unconstrained data sets for natural scene classification and on first-person life logging data?

- What types of visual features work best for life-logging indoor scene classification, and to what extent they can estimate the fine-grained location of life logging images?
- Can the temporal relationships between sequentially captured life logging images be modeled to improve the classification performance?
- To which extent can we detect potentially sensitive objects, in particular computer screens in lifelogging images?

The key contributions of this thesis are:

- We present image classification systems for unconstrained, realistic, large scale data sets that include methods for dealing with noisy and biased data in social and lifelogging images.
- We provide novel, realistic, and large-scale datasets for testing classification algorithms.
- We propose new applications in ecology domain, that are interesting in their own right and have the potential to give insight into how to accurately crowd-source other types of information from large, noisy social image sharing website.
- We propose novel applications in privacy domain to identify sensitive photo of first person camera users.

The work in this thesis is directly based on initial work published in the following papers:

- Haipeng Zhang, Mohammed Korayem, David Crandall, and Gretchen Lebuhn. *Mining photo-sharing websites to study ecological phenomena*. In Proceedings of the 21st International Conference on World Wide Web,pp. 49-758, ACM, 2012.
- Jingya Wang, Mohammed Korayem, and David Crandall. *Observing the natural world with Flickr*. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 452-459, IEEE, 2013.

- Robert Templeman, Mohammed Korayem, David Crandall, and Apu Kapadia. *PlaceAvoider: Steering first-person cameras away from sensitive spaces*. In Proceedings of the Network and Distributed System Security Symposium (NDSS). 2014.
- Mohammed Korayem, Robert Templeman, Dennis Chen, David Crandall, and Apu Kapadia. *ScreenAvoider: Protecting computer screens from ubiquitous cameras*. arXiv preprint arXiv:1412.0008 (2014).
- Mohammed Korayem, Abdallah Mohamed, David Crandall, and Roman Yampolskiy. *Learning visual features for the avatar captcha recognition challenge*. In Proceedings of the 11th International Conference on Machine Learning and Applications (ICMLA), pp. 584-587, IEEE, 2012.
- Mohammed Korayem, Abdallah Mohamed, David Crandall, and Roman Yampolskiy. *Solving avatar captchas automatically*. In Proceedings of the Advanced Machine Learning Technologies and Applications, Communications in Computer and Information Science Volume 322, pp. 102-110. Springer, 2012.

In addition, the thesis draws on work developed in other recent papers in related domains:

- Mohammed korayem, and David Crandall. *De-Anonymizing users across heterogeneous social computing platforms*. In Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media (ICWSM). 2013.
- Haipeng Zhang, Mohammed Korayem, Erkang You, and David Crandall. *Beyond co-occurrence: Discovering and visualizing tag relationships from geo-spatial and temporal similarities*. In Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM), pp. 33-42, ACM, 2012.
- Khalifeh Aljadda, Mohammed Korayem, Camilo Ortiz, Trey Grainger, John A. Miller, and William S. York. *PGMHD: A scalable probabilistic graphical model for massive hierar-*

chical data problems. In Proceedings of the IEEE International Conference on Big Data (Big Data), pp. 55-60, IEEE, 2014.

- Khalifeh Aljadda, Mohammed Korayem, Trey Grainger, and Chris Russell. *Crowd-sourced query augmentation through semantic discovery of domain-specific jargon*. In Proceedings of the IEEE International Conference on Big Data (Big Data), pp. 808-815, IEEE, 2014.
- Mohammed Korayem, David Crandall, and Muhammad Abdul-Mageed. *Subjectivity and sentiment analysis of arabic: A survey*. In Proceedings of the Advanced Machine Learning Technologies and Applications, Communications in Computer and Information Science Volume 322, pp. 128-139, Springer, 2012
- Muhammad Abdul-Mageed, Mona Diab, and Mohammed Korayem. *Subjectivity and sentiment analysis of modern standard arabic*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 587-591, 2011.
- Muhammad Abdul-Mageed, Mohammed Korayem, and Ahmed YoussefAgha . *"Yes we can?": Subjectivity annotation and tagging for the health domain*. In Proceedings of the Recent Advances in Natural Language Processing. 2011.

The rest of the thesis is organized as follows. In Chapter 2, we present an overview of image classification. We start by describing basic image classification concepts and components, and then we give an overview about Convolutional Neural Networks. In Chapter 3, we give a practical example of an image classification system that discriminates between synthetic and real faces. This simple but useful example demonstrates the potential unexpected consequences caused by dataset bias. Our system was able to solve an image-based Captcha recognition competition [78, 79]. We then move on to our major contributions in Chapter 4 and Chapter 5. In Chapter 4, we present a novel idea to mine photo-sharing

website to study ecology phenomena [139, 143]. We present a system to utilize image classification and deep learning to deal with noisy and biased data social media. In Chapter 5, we show our work towards maintaining privacy of first-person camera users through analyzing images. We present two systems in that direction, PlaceAvoider [128] and ObjectAvoider [80]. Finally, we conclude this thesis in Chapter 6.

CHAPTER 2

BACKGROUND ON IMAGE CLASSIFICATION

2.1 INTRODUCTION

Broadly described, image classification is process that assigns one or more predefined labels to an image based on visual content. Image classification differs from other vision problems (e.g, segmentation, localization, instance recognition, and category recognition). For example, the goal of segmentation is to divide the image into parts or segments, grouping pixels together based on a similarity measure, in that it does not try to recognize or label localized parts of an image. While image classification tries to give label or set of labels for the whole image, segmentation gives a label or labels for each pixel. Localization finds the position of an object inside the image. Image classification is thus on one hand simpler than other vision problems in that it requires predicting only single or multiple values per image, and is a concrete and well defined problem for testing vision features, classifiers, and algorithms. At the same time, it is more challenging than simple recognition – for instance, scene classification may involve recognizing multiple objects and understanding the relationship between them.

In practice, designing an image classification system typically requires four important steps: data set generation, visual feature development, classifier learning, and evaluation.

The approach for each component needs to be tailored to the requirements of each specific application. Dataset biases should be considered. A visual feature may perform well on some applications but not others. Classifier performance varies from task to task. And there are many possible evaluation metrics for any task. Of these, perhaps the most important design choice is selecting the appropriate visual features [27]. Section 2.2 describes each of these components in more detail. Section 2.3 gives an overview of deep learning, which is the most recent methodology for image classification. Finally, we summarize this chapter in Section 2.4.

2.2 IMAGE CLASSIFICATION COMPONENTS

2.2.1 DATASETS

The collection and selection of an appropriate training and testing datasets is foundational to every remaining component of an image classification system. For decades, the computer vision community has created benchmark datasets to evaluate methods in object recognition and image classification. These datasets cover a broad range of topics including whole scene recognition (SUN09 [142]), object detection (ImageNet [112], PASCAL [43], Caltech101 [45]), and object segmentation (LabelMe [113]). Figure 2.1 shows examples from the Caltech and ImageNet data sets.

The main objective for creating these data sets is to be representative of the visual world, however they end up as closed worlds with their own biases [132]. In addition to inherent biases, these datasets are often ill-posed for specific real world problems, containing irrelevant classes or unlikely viewpoints for the application. For instance, Figure 2.1 shows how objects in Caltech dataset are unrealistically cropped and centered, and included classes like soccer ball, Cartman, and images of brains that are not very important

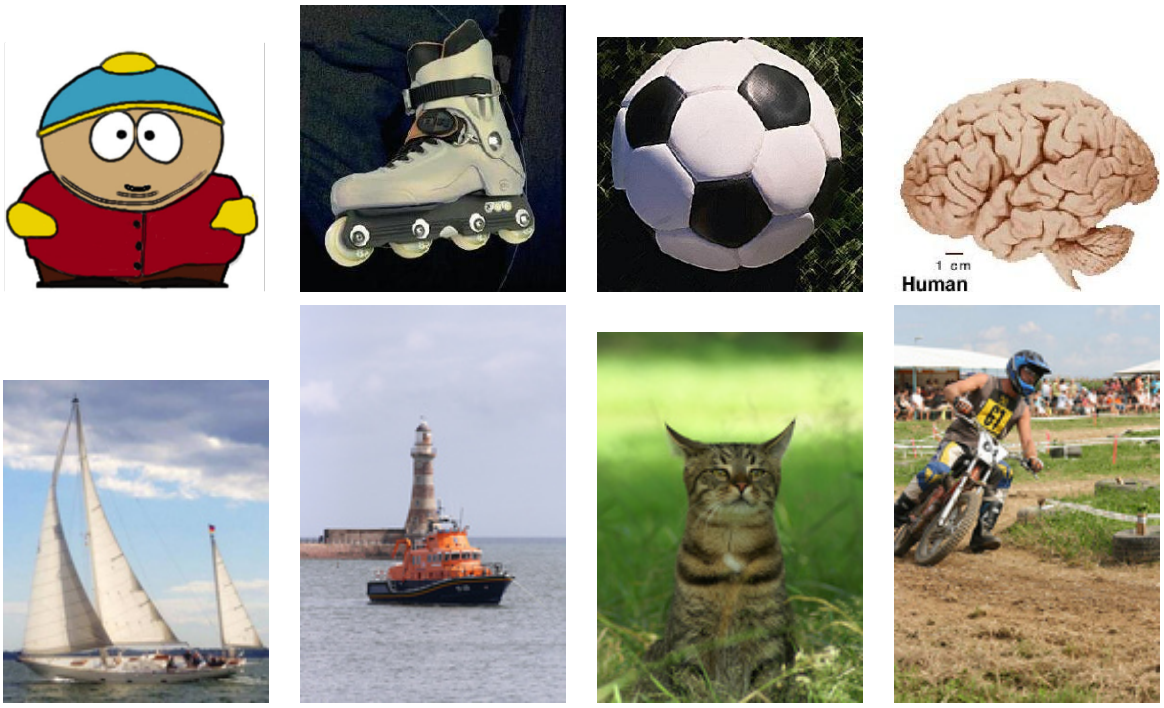


Figure 2.1: Sample image from Caltech dataset (top) and ImageNet dataset (bottom).

or common in the real world. It is difficult to imagine realistic applications that would involve these categories or images as clean as these. As shown in Figure 1.3, real consumer images are much more complicated and real applications would include much more complicated classifications tasks.

2.2.2 VISUAL FEATURES

Visual feature extraction is the second and arguably most influential component to the success of an image classification system. Over the years, the vision community has developed a range of standard features which can be broadly grouped by the information that they are designed to capture. Many features are based on color such as simple histograms in RGB, HSV, or CIELAB color spaces [142]. Others try to capture and quantify local textures such as Local Binary Pattern (LBP) [141] and Maximum Response filters (MR8) fea-

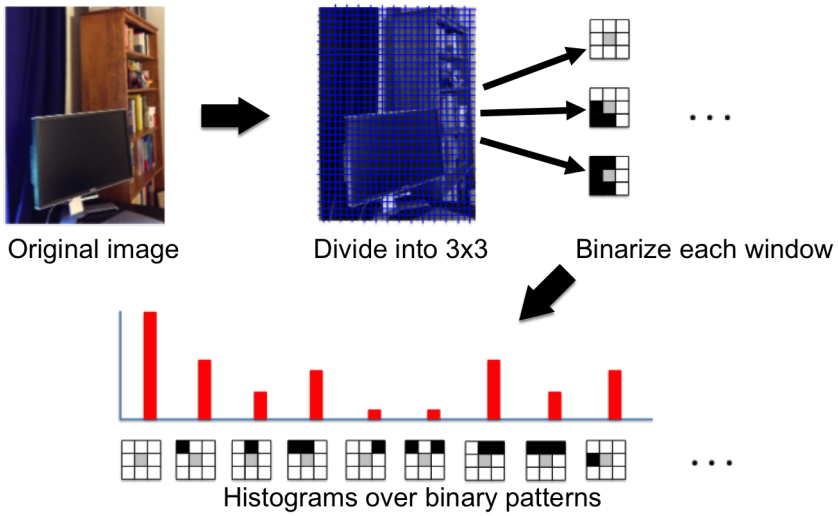


Figure 2.2: Example for extraction Local Binary Pattern feature

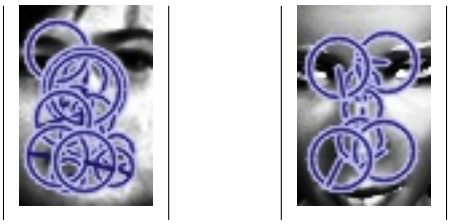


Figure 2.3: Detected SURF features for a human face (left) and avatar face (right).

tures [137]. There are also features which take advantage of image gradients (edges) to describe both global and local scene structure, such as GIST [107] and Histograms of Oriented Gradients (HOG) [37] respectively. Here, we give brief descriptions of some of the most popular features:

- **Summary statistics**: The simplest features compute summary statistics about an image's raw pixel data (e.g., maximum, minimum, mean, median, and sum of the pixel values).

- **Histograms:** A slightly more sophisticated feature, histograms can be computed in

gray scale or color at different bin sizes. They are very simple to compute, but are weak representations of an image – all the spatial structure of the image is lost, for example.

- **Histograms of Oriented Gradients (HOG):** HOG is a very popular feature extraction technique for recognizing objects including humans [37]. The idea is to break an image into a grid of small windows, compute edge strengths and directions, and then compute a weighted histogram of edge orientations within each window. HOG features capture the overall shape of an object or image region, but give invariance to illumination and contrast changes, and allow for some variation in shape and appearance. Figure 2.4 shows an example of the pipeline to extract HOG features.

- **GIST descriptors:** GIST features [107] try to capture the overall appearance ("gist") of a scene. To do this, the image is divided into a grid of non-overlapping cells, and color and texture features inside each cell are computed. These features are concatenated together to produce a single feature vector for each image. GIST is invariant or insensitive to a variety of image transformations including illumination changes, blur, and resizing, but is not invariant to translation, rotation, etc.

- **Local binary pattern (LBP):** The local binary pattern (LBP) [141] descriptor examines each pixel in a small neighborhood of a given pixel, and assigns a binary bit depending on whether the grayscale value is greater than or less than that of the central pixel. The bits that represent the comparison are then concatenated to form an 8-bit decimal number, and a histogram of these values is computed over all windows of

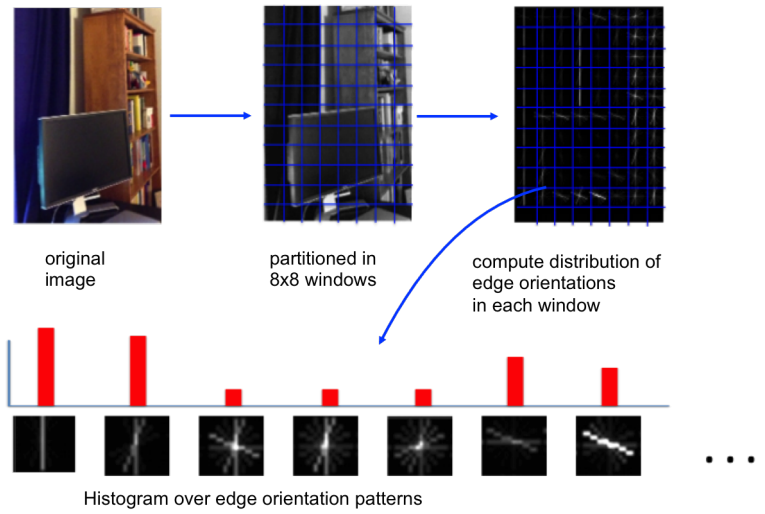


Figure 2.4: Example for extraction HOG feature

the image. Figures 2.2 shows an example of the pipeline to extract the LBP features.

- **Quantized feature descriptors:** Another popular technique is to detect a sparse set of highly distinctive *feature points* in an image, calculate an invariant descriptor for each point, and then represent an image in terms of a histogram of vector-quantized descriptors [36]. The most common examples for invariant descriptor are the Scale-Invariant Feature Transform (SIFT) [96] and Speeded-Up Robust Features (SURF) [16]. Figure 2.3 shows an example for extracted SURF features.

Regardless of the specific technique, all of these features share the common property that they take an image as input and produce a high-dimensional feature vector as output.

2.2.3 CLASSIFIERS

Given the dataset and extracted visual features, the problem of image classification is reduced to a high-dimensional classification problem. There is a wealth of literature and technical developments within both the vision and machine learning communities sup-

porting this sort of classification.

Popular techniques include linear classifiers such as L2-regularized logistic regression [44], max-margin methods; for example, Support Vector Machines (SVMs) [25], non-linear probabilistic models such as Naïve Bayes [57, 71], instance-based classifiers such as K nearest neighbor [62], and neural network models; for instance, feed forward neural networks [58]. These classifiers have different assumptions; for example, linear classifiers assume that the data is linearly separable or can be transformed to space where it will be linearly separable, and Bayesian classifiers assume strong independence assumptions between features. Given a problem description, it is often unclear which method will perform optimally and in practice their relative performance can change depending on the task, so developing a new vision system typically involves exhaustive experimentation to choose and customize the learning algorithm for a specific application.

2.2.4 EVALUATION

The final component of any classification pipeline is evaluation. Performance can be measured in a number of different ways and different applications emphasize different metrics that quantify different qualities. Accuracy is a widespread and easy-to-understand measure of system performance defined as the fraction of correct predictions over the number of data points. However, it is misleading or difficult to interpret with skewed class distributions (i.e. a task with nine times more positive examples could achieve 90% accuracy simply by declaring everything to be positive). True Positive Rate (TPR) measures the proportion of positive exemplars that are correctly identified as positive, while True Negative Rate (TNR) measures the proportion of negative exemplars that are correctly identified as negative. A Receiver operating characteristic (ROC) curve is a graphical plot presenting

the relationship between true positive rate and false positive rate.

Other metrics are designed to capture the retrieval performance of a classifier relative to a specific class. Precision measures the proportion of correctly predicted examples of the class over the total number of predictions of that class. Recall measures the proportion of exemplars of a specific class that are correctly identified over the total number of exemplars of that class. Applications that require few positive exemplars to be misclassified as negative would prefer high recall over high precision: an example might be a preliminary medical test that can be easily verified afterwards. Others may prefer high precision over high recall; e.g., in image search it might be better to retrieve a small set of accurate results than a larger one with non-related results.

2.3 DEEP LEARNING FOR IMAGE CLASSIFICATION

In the previous section, we described the traditional image classification system that has been popular for most of the last two decades. Perhaps the most important design choice for this framework is the visual feature, which are designed by hand. Intuitively, no single feature performed best across all tasks and the feature selection process involved many experiments and much hand tuning. In this section, we present a recent innovation in the field that is capable of learning the correct feature representation for a given task automatically.

Recently, the Convolutional Neural Network (CNN) [82] has gained a lot of attention in the vision community, ever since it outperformed all other techniques in the ImageNet challenge in 2012 (the most famous object category detection competition) [112]. CNN is a special type of feed forward neural network inspired by the biological process [82] in cats' visual cortex. CNN enjoys additional features that distinguish it from standard

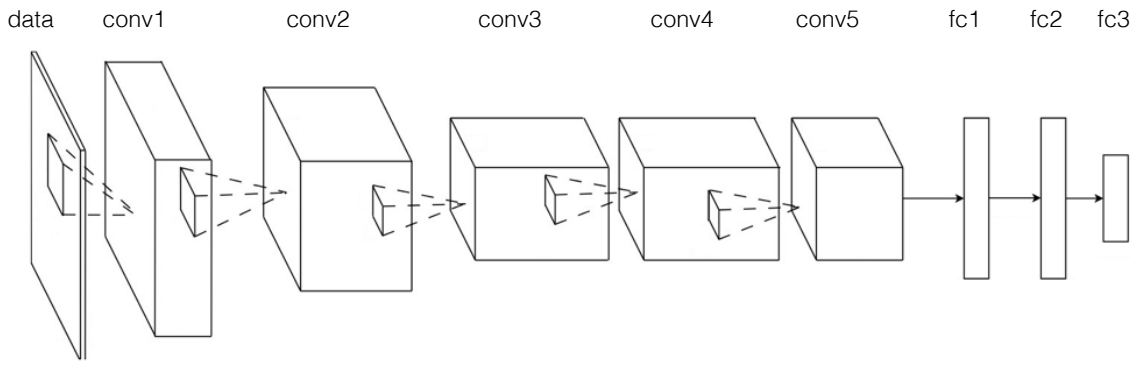


Figure 2.5: An example of a deep learning network. The sparsely connected convolutional layers (conv1, conv2, conv3, conv4, conv5) extract image features while the fully connected nodes (fc1,fc2,fc3) express the model in terms of features. This structure's for the ImageNet network model in [69]. Diagram modified from the original one in [69])

neural networks: shared weights and sparse connectivity. A layer in a CNN may consist of three different stages: convolution, non-linear activation, and pooling. In the convolution stage, a set of convolution filters is applied in parallel. The output of a convolution filter is then passed to non-linear activation functions (e.g., a rectified linear function or sigmoid function). The final stage is pooling, where the network output is manipulated based on its neighbors (e.g., max pooling, L_2 norm, and weighted average). Pooling makes the network invariant to the translation of the input. Sometimes these three stages are created as three layers in the network structure (e.g, in the CAFFE framework [69]). An example of a CNN network is shown in Figure 2.5. The main interesting idea behind this approach is to learn the image representation (visual features) and a classifier at the same time, instead of first designing hand-crafted low-level features, and then applying a machine learning algorithm.

A major issue of CNN models is the need for large training data sets, because the depth of the networks require millions of images to estimate millions of parameters; otherwise the network will overfit. Recent work has shown a way around this problem: with insuffi-

cient training data sets, one solution involves using models that are trained on very large data sets, then using these models as initial weights for networks that need to be trained on relatively small data sets. It is not clear why this approach works, but one interpretation is that the network that is trained on very large scale data set learns some general low-level features about the visual world, and then is retrained or "fine tuned" to learn some high level representation and organization of these features on the specified data set.

In the following chapters we will show how we incorporate CNNs within our system for scientific and privacy applications. The main objective of applying the CNN is to study how well deep learning approaches perform on real vision problems outside the standard data sets. Despite their success, deep learning has several drawbacks that we do not understand well. It is still not clear why CNNs perform well on some but not all problems. Moreover, they can learn uninterpretable solutions that sometimes have highly counter-intuitive properties [126], for example, they are dramatically misclassifying exemplars that are imperceptibly different from one another.

2.4 SUMMARY

To conclude this chapter, we presented the main components for traditional image classification system: data set generation, visual feature development, classifier learning, and evaluation. Then, we described deep learning with the Convolutional Neural Network approach for image classification. In the following chapter, we illustrate these components through a prototypical image classification system to differentiate between real and simulated images of human faces. This straightforward binary classification system is key to solving a publicly available security challenge.

CHAPTER 3

AN EXAMPLE OF IMAGE CLASSIFICATION METHODS AND PITFALLS : AVATAR CAPTCHA RECOGNITION CHALLENGE

To illustrate the approach, potential pitfalls, of typical image classification work, in the this chapter we consider a problem in which the system is tasked to differentiate between real and simulated images of human faces. This simple but useful example also shows the potential unexpected consequences caused by bias of the data. This binary classification problem is key to solving a publicly available security challenge. Souza et. al. proposed that the ability of humans to distinguish real and synthetic faces far exceeded automatic systems and could be used to differentiate between human users and automated agents ("bots") [42]. The system presented users with a set of twelve images and asked them to label each image as real or synthetic. It is a well defined problem with limited and clean images. The system's designers found that humans were able to correctly identify all human faces a little under two-thirds of the time, while randomly guessing the answers would succeed fewer than one out of five thousand attempts. Authors presented their system as a secure technique for identifying bots, and challenged the vision community to design an automatic procedure that could outperform humans. We show that this task is surprisingly easy to solve in an image classification framework, with the results not only matching but *far outperforming* human accuracy.

3.1 BACKGROUND

Online activities play an important role in our daily life, allowing us to carry out a wide variety of important day-to-day tasks including communication, commerce, banking, and voting [9, 49]. Unfortunately, these online services are often misused by undesirable automated programs, or "bots," that abuse services by posing as human beings to (for example) repeatedly vote in a poll, add spam to online message boards, or open thousands of email accounts for various nefarious purposes. One approach to prevent such misuse has been the introduction of online security systems called Captchas, or Completely Automated Public Turing tests to tell Computers and Humans Apart [9]. Captchas are simple challenge-response tests that are generated and graded by computers, and that are designed to be easily solvable by humans but beyond the capabilities of current computer programs [140]. If a correct solution for a test is received, it is assumed that a human user (and not a bot) is requesting an Internet service. There are three main categories of Captchas, as presented in [24]: text-based, sound-based, and image-based Captchas. Other work has combined some of these categories into multi-modal Captchas [11].

The strength of a Captcha system can be measured by how many trials an attacking bot needs on average before solving it correctly [24]. However, there is a tension between developing a task that is as difficult as possible for a bot, but is still easily solvable by human beings. This is complicated by human users who may have sensory or cognitive handicaps that prevent them from solving certain Captchas. The best Captcha schemes are thus easy for almost any human but almost impossible for an automated program to solve.

3.1.1 AVATAR CAPTCHA RECOGNITION CHALLENGE

Recently, a novel image-based system was proposed called *Avatar Captcha* [42] in which users are asked to perform a face classification task. In particular, the system presents a set of face images, some of which are actual human faces while others are avatar faces generated by a computer, and the user is required to select the real faces. The designers of the scheme found that humans were able to solve the puzzle (by correctly finding all human faces) about 63% of the time, while a bot that randomly guesses the answers would pass only about 0.02% of the time.

In this chapter [78, 79], we consider how well a bot could perform against the Avatar Captcha if it used computer vision algorithms instead of random guessing. We test a variety of modern learning-based recognition algorithms to classify human and avatar face images released by the authors of the challenge [42]. Through these experiments we found that this task is surprisingly easy, with some algorithms actually *outperforming* humans on this dataset. Our results show that this captcha is not as secure as the authors had hoped; however, our analysis suggests that the problem may not be in the idea itself but rather in the way the data was generated. The algorithmic way that the images were generated allows the recognition algorithms to learn subtle biases in the data.

3.2 DATASETS

We used the dataset released by the authors of the Avatar Captcha system [42] as part of the ICMLA Face Recognition Challenge. This set consists of grayscale 100 photos, evenly split between human and avatar faces. The faces are all generally frontal-view with some variation in illumination, facial expression, and background. The human images come from the Nottingham scans dataset and consist of real images of men and women, and are

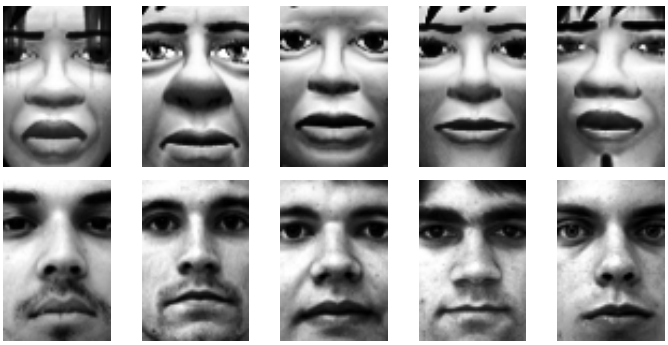


Figure 3.1: Sample avatar (top) and human faces (bottom) from our dataset.

resized to a common resolution of 50×75. The avatar images are a sample of avatar faces from the Entropia Universe virtual world, and were also resized to a resolution of 50×75. Figure 3.1 presents examples of different facial images from this dataset.

3.3 VISUAL FEATURES

We used a wide variety of visual features, including those introduced in Section 2.2. The details of our features are:

- **Naïve features:**
  - **Summary statistics**: Our simplest features compute summary statistics about an image. We tried a 1-dimensional feature that is simply the mean pixel value of the image, and a 5-dimensional feature including maximum, minimum, mean, median, and sum of the pixel values.
  - **Grayscale histograms:** We computed grayscale histograms for each image. We tried histograms with 2, 4, 8, 16, 32, 64, and 128 bins.
  - **Vectors of raw pixel values:** This feature involves simply reshaping an image into a vector by concatenating all of the image rows of grayscale values together. The resulting feature vector has $50 \times 75 = 3750$ dimensions.

- **Histograms of Oriented Gradients (HOG):** We compute HOG to form a single $2,268$ dimensional feature vector [37].
- **GIST descriptors:** We use GIST features [107] to capture the overall appearance ("gist") of a scene. Our GIST uses a 4×4 grid and computes 60 features per cell, yielding a 960 dimensional vector for our images.
- **Quantized feature descriptors:** We use SURF to detect features points and calculate descriptors for each point, and then use k-means to produce a set of 50 clusters or "visual words." We then assign each descriptor to the nearest visual word, and represent each image as a histogram over these visual words, yielding a 50 dimensional feature vector. Figure 2.3 illustrates some detected SURF features.
- **Local binary pattern-based features:**
  - **Four Patch Local Binary Pattern (FPLBP)** is an extension to the original LBP where for each pixel in the image we consider two rings, an inner ring of radius r_1 and an outer one of radius r_2 (we use 4 and 5, respectively), each centered around a pixel [141]. T patches of size $s \times s$ (we use $s = 3$) are spread out evenly on each ring. Since we have T patches along each ring then we have $T/2$ center symmetric pairs. Two center symmetric patches in the inner ring are compared with two center symmetric patches in the outer ring, each time setting one bit in each pixel's code based on which of the two pairs are more similar, and then calculate a histogram from the resulting decimal values.
  - **Local Difference Pattern Descriptor:** We also introduce a simple modification to the Local Binary Pattern descriptor (LBP) which we call Local Difference Pattern. We divide the image into n x n $(3$ x $3)$ windows and compute a new value for the center of each window based on the values of its neighbors. We compute the new value as the average of the differences between the center and all other

original image | LBP | LDP | LDP-absolute difference

Figure 3.2: Illustration of LBP and LDP features for a human face.

pixels in the window (instead of computing the binary window and converting it into its decimal value as in LBP). We tried using both absolute and signed differences. Figure 3.2 illustrates this feature. Finally we compute a histogram for these new values.

3.4 CLASSIFIERS AND FEATURE SELECTION METHODS

We learned two different types of classifiers: Naïve Bayes [57,71], and L2-regularized logistic regression [44]. One is an example of a non-linear classifier and the other is an example of a linear classifier. Naïve Bayes (NB) is a non-linear simple probabilistic classifier, which makes a prediction using the equation

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}}(P(y) \prod_{j=1}^{n} P(d_j|y)), \tag{3.1}$$

where $\hat{y}$ is predicted class, Y is the set of possible class labels, d is the feature vector, and d_j is the j^{th} dimension of feature vector. L2-regularized logistic regression is a linear classifier which makes decisions based on $sign(w^T d)$, where the class is positive when $w^T d > 0$, and negative otherwise (w is the weight vector). To find the w, the learning algorithm solves

the following optimization problem:

$$\hat{w} = \underset{w}{\operatorname{argmin}} \left\{ \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log \left(1 + e^{-y_i w^T d_i} \right) \right\}, \tag{3.2}$$

where C is a regularization parameter, d_i is the i^{th} data point (vector), y_i is the label for the i^{th} data point, and n is the total number of data points. We used Correlation-based Feature Selection (CFS) [56] to reduce feature dimensionality. CFS mainly depends on the following assumption: good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other [56].

3.5 EVALUATION OF THE CLASSIFICATION SYSTEM

We evaluated the performance of various combinations of the above visual features, classifiers, and feature selection algorithms on the Avatar Captcha recognition task. Table 3.1 shows the results for our simplest features with a Naive Bayes classifier. The table shows results on both the two-class task of deciding if a given single facial image is an avatar or a human, and the full Avatar Captcha task in which of 12 images must all be classified correctly. All experiments in this section were conducted using 10-fold cross-validation. The best 2-way classification accuracy in this set of experiments was 93% when raw pixel values were used. This means that an automated program could correctly answer an Avatar Captcha with probability 41.9%, or nearly 2,000 times more accurately than predicted by [42]. The histogram-based techniques also achieve relatively good classification performance, with 89% accuracy for 256-bin histograms and 92% for 128-bin histograms. Even the simplest feature (1-d feature consisting of average pixel value) performs nearly 7 percentage points better than baseline.

Table 3.1: Experimental results with simple features and Naive Bayes classifiers, on classifying a single image as well as the 12-way Avatar Captcha task.

Method	2-class accuracy	Captcha accuracy
Pixel values	**93%**	**41.9%**
Mean pixel	57%	0.1%
Summary stats (mean, median, min, max, sum)	61%	0.3%
Histograms (256-Bins)	89%	24.7%
Histograms (128-Bins)	92%	36.8%
Histograms (64-Bins)	77%	4.3%
Histograms (32-Bins)	78%	5.1%
Histograms (16-Bins)	75%	3.2%
Histograms (8-Bins)	77%	4.3%
Histograms (4-Bins)	69%	1.2%
Histograms (2-Bins)	52%	0.03%
Random baseline	50%	0.02%

Table 3.2 shows results with the more sophisticated image features and classifiers. Surprisingly, we actually achieve perfect classification (100% accuracy) on the test dataset when the high-dimensional feature vector of raw pixel values is combined with the LibLinear classifier. According to this result, an automated bot could successfully solve Avatar Captchas correctly with nearly perfect accuracy, performing even better than humans on this task! HOG features achieved 99% accuracy with LibLinear. The table also shows that feature selection could successfully reduce the dimensionality of the feature vectors while sacrificing little performance, since the 54-dimensional reduced vectors for raw pixel values achieves 98% accuracy with Naive Bayes.

The surprisingly high performance of relatively simple vision algorithms on the Avatar Captcha task suggests that there may be biases in the dataset that are readily discovered

Table 3.2: Experimental results with more sophisticated features and classifiers, including LibLinear, Naive Bayes, and Naive Bayes with feature selection. Feature dimensionality is shown inside parentheses.

Method	LibLinear	Naive Bayes (NB)	NB+FS
Pixel values	**100% (3750d)**	93% (3750d)	98% (54d)
256-bin Histogram	60% (256d)	**89% (256d)**	82% (24d)
GIST	84% (960d)	88% (960d)	**90% (24d)**
HOG	**99% (2268d)**	94% (2268d)	95% (44d)
FPLBP	94%(240d)	89%(240d)	**95%(26d)**
SURF codebook	**97% (50d)**	96% (50d)	94% (22d)
LDP (absolute differences)	94% (256d)	99% (256d)	**100% (61d)**
LDP (differences)	96% (256d)	**98% (256d)**	99% (75d)
LBP	**98% (256f)**	95% (256f)	**98% (31f)**

and exploited by machine learning. For example, the fact that a classification algorithm looking only at mean pixel value achieved a significant improvement over baseline indicates that the images in one class are on average brighter than those of the other class. Other more subtle biases likely exist, since the sets of facial images were generated in two very different ways (one through photography and the other with computer graphics).

We did some investigation to test whether applying some simple transformations to images could make the problem more difficult and thus confound the classification algorithms. In particular, we tried three types of transformations:

- **Noise:** We tried adding different types of random noise to the images, including Gaussian, Poisson, and Salt & Pepper noise. For each image, we randomly chose one type of noise and then added it to the image.
- **Rotation:** To increase the appearance variation in the dataset, we tried rotating each image by a random angle between 1 and 180 degrees.
- **Occlusion:** Finally, we tried to explicitly defeat the classification algorithms by iden-

Table 3.3: Classification performance on images corrupted by noise, rotations, and occlusions.

	Classifier	Pixel values	Histogram	Gist	HOG
Original	Naive Bayes + Feature Selection	98%	82%	90%	95%
	LibLinear	100%	60%	84%	99%
Noise	Naive Bayes + Feature Selection	98%	46%	89%	94%
	LibLinear	100%	61%	84%	90%
Rotation	Naive Bayes + Feature Selection	86%	74%	66%	81%
	LibLinear	93%	92%	65%	81%
Occlusion	Naive Bayes + Feature Selection	91%	86%	91%	99%
	LibLinear	99%	90%	89%	97%

tifying the 500 most important pixel locations in the image (by looking at the top features identified by feature selection on the raw pixel vectors), and occluding them by setting them to 0.

Table 3.3 shows the results for the different features and classifiers when applied to datasets that have been corrupted by the above techniques. Adding random noise successfully confounds the histogram features, reducing accuracy from 82% to near the random baseline, but has little effect on other features. Rotations confuse HOG, GIST and pixel vectors since these features encode spatial position explicitly, but have minimal effect on histogram features. The occlusion features reduce performance significantly for Naive Bayes with feature selection, but have little impact otherwise. Combining all three techniques together reduces the accuracy of the best-performing classifier from 100% down to 85%. In the Avatar Captcha task, in which 12 images must be correctly classified, this accuracy means that even with the noisy images a bot could solve the problems about 14% of time. Figure 3.3 shows some sample images corrupted by rotation and noise.

Figure 3.3: Avatar (top) and human faces (bottom) after noise and rotation.

3.6 SUMMARY

To conclude, we have applied a variety of visual features and classifiers to the problem of distinguishing between human and avatar faces. Our results show that while automated bots are very unlikely to solve Avatar Captchas through random guessing, computer vision techniques can solve these tasks substantially better than humans. We suspect that the high performance may be enabled by subtle differences and biases between the avatar and face images in the ICMLA Face Recognition Challenge dataset. While Avatar Captchas may have advantages in usability, our results demonstrate that in practice it is likely to be very difficult to secure them against attacks based on modern computer vision and machine learning techniques. This work serves as both a case study of how to design image classification techniques to a practical problem, but is also a cautionary tale: dataset bias can be subtle but extremely problematic, leading to high unexpected results.

CHAPTER 4

OBSERVING THE NATURAL WORLD THROUGH PHOTO-SHARING WEBSITES

4.1 INTRODUCTION

In the previous chapters, we covered image classification concepts and methods, and then we illustrated these concepts through a practical example. Now, we present our work towards determining if observations of natural phenomenon can be mined from large-scale, user-generated image collections. We present a system to mine photo-sharing web site to study ecology phenomena at large scale using visual and textual features. This differs significantly from that presented in the previous chapter both in scale of the dataset as well as its composition; unlike in the previous problem, the images here are not well aligned and vary drastically in image content.

The popularity of social media websites, such as Flickr and Twitter, has generated huge collections of user-generated content online. Billions of photos are posted on these websites, forming a massive social sensor network that captures the visual world. While a tweet is a textual expression of the state of a person and his or her surroundings, a photo is a visual snapshot of what the world looked like at a certain point in time. For example, an outdoor image contains information about the state of the natural world, such as weather conditions and the presence or absence of plants and animals. The billions of images on so-

cial media websites could be analyzed to recognize these natural objects and phenomena, creating a new source of data to biologists and ecologists.

There are two key challenges to mine the ecological information latent in these photo datasets. The first is how to recognize ecological phenomena appearing in photos and how to map these observations to specific places and times and the second is how to deal with the biases and noise inherent in online data.

Our proposed methods handle these two key challenges through image classification (to detect the ecological phenomena appearing in photos) and probabilistic models to deal with biases and noise inherent in online data. As an example, Figure 4.1 shows the resulting map produced by our automated Flickr analysis, and compares it to the corresponding snow cover map produced by NASA's MODIS instrument [55]. We note that the Flickr map is much sparser than the satellite map, especially in sparsely populated areas like northern Canada and the western U.S. On the other hand, the Flickr maps give some observations even when the satellite maps are missing data due to clouds.

In this chapter [139, 143], we introduce the novel idea of mining photo-sharing sites for geo-temporal information about the natural world (e.g, ecological phenomena). We present a framework based on image classification and probabilistic models for deriving crowd-sourced observations from noisy, biased data using both visual and textual tag analysis. We test our hypothesis by recognizing specific types of scenes and objects in large-scale image collections from Flickr. We consider a well-defined but nevertheless interesting problem: (1) deciding whether there was snow on the ground at a particular place and on a particular day, given the set of publicly-available Flickr photos that were geo-tagged and time-stamped at that place and time, and (2) Estimation of vegetation cover.

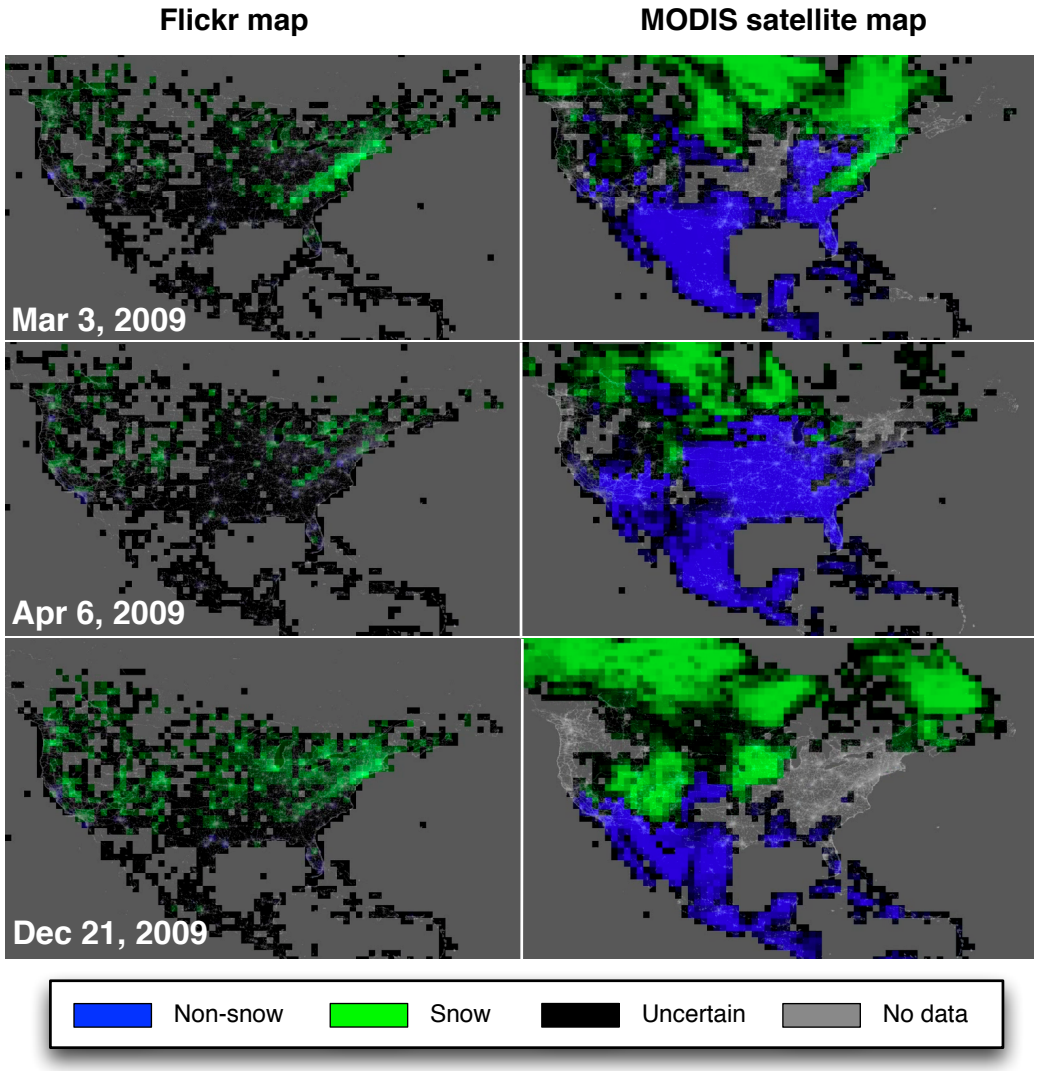


Figure 4.1: Automatically-generated snow cover maps generated by our Flickr analysis (left), compared with satellite maps (right), on three days. Green indicates snow, blue indicates no snow, and gray indicates uncertainty (caused by too few photos in Flickr analysis, or by cloud cover in satellite maps).

4.2 RELATED WORK

A variety of recent work has studied how to apply computational techniques to analyze online social datasets in order to aid research in many disciplines [87]. Much of this work has studied questions in sociology and human interaction, such as how friendships form [32], how information flows through social networks [95], how people move through space [22], and how people influence their peers [12]. The goal of these projects is not to measure data about the physical world itself, but instead to discover interesting properties of human behavior using social networking sites as a convenient data source.

Crowd-sourced observational data. Other studies have shown the power of social networking sites as a source of observational data about the world itself. Bollen *et al.* [18] use data from Twitter to try to measure the aggregated emotional state of humanity, computing mood across six dimensions according to a standard psychological test. Intriguingly, they find that these changing mood states correlate well with the Dow Jones Industrial Average, allowing stock market moves to be predicted up to 3 days in advance. However their test dataset is relatively small, consisting of only three weeks of trading data. Jin *et al.* [70] use Flickr as a source of data for prediction, but they estimate the adoption rate of consumer photos by monitoring the frequency of tag use over time. They find that the volume of Flickr tags is correlated with with sales of two products, Macs and iPods. They also estimate geo-temporal distributions of these sales over time but do not compare to ground truth, so it is unclear how accurate these estimates are. In contrast, we evaluate our techniques against a large ground truth dataset, where the task is to accurately predict the distribution of a phenomenon (e.g., snow) across an entire continent each day for several years.

Crowd-sourcing from social media. Several recent studies have shown the power of social media for observing the world itself, as a special case of 'social sensing' Aggarwal *et al.* [8]. This work includes using Twitter data to measure collective emotional state [52] (which, in turn, has found to be predictive of stock moves [18]), predicting product adoption rates and political election outcomes [70], and collecting data about earthquakes and other natural disasters [116].

Particularly striking examples include Ginsberg *et al* [50], who show that geo-temporal properties of web search queries can predict the spread of flu, and Sadilek *et al* [114] who show that Twitter feeds can predict when a given person will fall ill. A recent exciting project, Yahoo Weather [1], reflects the big potential of mining Flickr images. From images uploaded by customer all over the world, the intelligent system picks one high quality photo that best represents the weather (cloudy, stormy or snowy) at each location and time period.

Crowd-sourced geo-temporal data. Other work has used online data to predict geo-temporal distributions, but again in domains other than ecology. DeLongueville *et al.* [39] study tweets related to a major fire in France, but their analysis is at a very small scale (a few dozen tweets) and their focus is more on human reactions to the fire as opposed to using these tweets to estimate the fire's position and severity. In perhaps the most related existing work to ours, Singh *et al.* [123] create geospatial heat maps (dubbed "social pixels") of various tags, including snow and greenery, but their focus is on developing a formal database-style algebra for describing queries on these systems and for creating visualizations. They do not consider how to produce accurate predictions from these visualizations, nor do they compare to any ground truth.

Accuracy of geo and temporal data on Flickr. Over a sample of 10 million images on

Flickr.com, 37% of them probably have incorrect timestamps [130]. The accuracy of geo-location is limited due to the camera device, and GPS precision.

Meanwhile, a lot of work is trying to correct estimate or correct geo-location of Flickr images. Singh *et al.* [122] estimates where images are taken for those missing geo-tags, by optimizing a graph clustering problem. Attributes in their graph include textual tags, timestamps and vision content. It is inspired by an earlier work by Crandall *et al.* [35]. Thomee *et al.* [130] show a detailed analysis of disagreement of camera time and GPS time. They also estimate a more accurate timestamp when users take multiple images in a short timespan. Hauff *et al.* [59] consider textual meta data to correct geotags. They also found for users active on both Flickr and Twitter, that a Twitter post at around the same time an image is taken can be a reliable reference to estimate the approximate location.

The specific application we consider here is inferring information about the state of the natural world from social media. Existing work has analyzed textual content, including text tags and Twitter feeds, in order to do this. Hyvarinen and Saltikoff [66] use tag search on Flickr to validate metereological satellite observations, although the analysis is done by hand. Singh *et al* [123] visualize geospatial distributions of photos tagged "snow" as an example of their Social Pixels framework, but they study the database theory needed to perform this analysis and do not consider the prediction problem.

Few papers have used actual image content analysis as we do here. Leung and Newsam [90] use scene analysis in geo-tagged photos to infer land cover and land use types. Murdock *et al* [101] analyze geo-referenced stationary webcam feeds to estimate cloud cover on a day-by-day basis, and then use these estimates to recreate satellite cloud cover maps. Webcams offer a complimentary data source to the social media images we consider here: on one hand, analyzing webcam data is made easier by the fact that the camera is stationary

and offers dense temporal resolution; on the other hand, their observations are restricted to where public webcams exist, whereas photos on social media sites offer a potentially much denser spatial sampling of the world.

We note that these applications are related to citizen science projects where volunteers across a wide geographic area send in observations [2], Fink *et al.* [46], King *et al.* [76]. These projects often use social media, but require observations to be made explicitly, whereas in our work we "passively" analyze social media feeds generated by untrained and unwitting individuals.

Detecting snow in images. We know of only a handful of papers that have explicitly considered snow detection in images. Perhaps the most relevant is the 2003 work of Singhal *et al* [124], Luo *et al.* [98] which studies this in the context of detecting "materials" such as water, grass, sky, etc. They calculate local color and texture features at each pixel, and then compute a probability distribution over the materials at each pixel using a neural network. They partition the image into segments by thresholding these belief values, and assign a label to each segment with a probabilistic framework that considers both the beliefs and simple contextual information like relative location. They find that sky and grass are relatively easy to classify, while snow and water are most difficult. Follow-up work Boutell *et al.* [21], Boutell [20] applied more modern techniques like support vector machines. Barnum *et al* [15] detect falling snow and rain, a complementary problem to the one we study here of detecting fallen snow.

Scene classification. Papers in the scene recognition literature have considered snowy scenes amongst their scene categories; for instance, Li *et al* [91, 92] mention snow as one possible component of their scene parsing framework, but do not present experimental results. The SUN database of Xiao *et al* [142] includes several snow-related classes like

"snowfield," "ski slope," "ice shelf," and "mountain snowy," but other categories like "residential neighborhood" sometimes have snow and sometimes do not, such that detecting these scenes alone is not sufficient for our purposes.

As we discussed in Section 2.3, a big advance in scene classification and object recognition has been achieved through deep learning (Convolutional Neural Networks (CNNs)) [26,53,82,144]. A CNN outperformed all other techniques in the ImageNet 2012 challenge (the most famous object category detection competition) [112]. The main advantage for deep learning is to learn image representation (visual features) and a classifier at the same time, instead of first designing hand-crafted low-level features, then applying a machine learning algorithm.

Vegetation classification. Kumar *et al.* [83] identifies plant species by leaf images. They focus on accurate leaf segmentation according to color difference of leaf and background, curvature distribution over scale, and nearest neighbor matching. Siagian *et al.* [121] introduces multiple Gist models in scene classification. There happens to be a test set of vegetation, and results show Gist features work well on vegetation classification.

The paper of Balamurugan *et al.* [14] is the closest one to our purpose. They consider color and texture features in images and obtain good results on detecting green vs non-green images. But they only test on a very limited dataset where the positive images are either with one tree in the center or full of trees or meadows. This is inadequate when we are working with very large number of publicly shared images.

The visual data in online social networking sites provide a unique resource for tracking biological phenomena: because they are images, this data can be verified in ways that simple text cannot. In addition, the rapidly expanding quantity of online images with geo-spatial and temporal metadata creates a fine-scale record of what is happening across

the globe. However, to unlock the latent information in these vast photo collections, we need mining and recognition tools that can efficiently process large numbers of images, and robust statistical models that can handle incomplete and incorrect observations.

4.3 METHODS

In this section we describe our framework for mining photo-sharing website. It consists of two main components: image classification and a probabilistic model. We start by describing our data sets, and then we show how we utilize image classification and deep learning to infer semantic from images using tags and visual features. Finally we present our probabilistic method to combine evidence from the image classification to improve our predictions.

4.3.1 DATASET

We use a sample of more than 200 million geo-tagged, timestamped Flickr photos as our source of user-contributed observational data about the world. We collected this data using the public Flickr API, by repeatedly searching for photos within random time periods and geo-spatial regions, until the entire globe and all days between January 1, 2007 and December 31, 2010 had been covered. We applied filters to remove blatantly inaccurate metadata, in particular removing photos with geotag precision less than about city-scale (as reported by Flickr), and photos whose upload timestamp is the same as the EXIF camera timestamp (which usually means that the camera timestamp was missing).

For ground truth we use large-scale data originating from two independent sources: ground-based weather stations, and aerial observations from satellites. For the ground-based observations, we use publicly-available daily snowfall and snow depth observations

from the U.S. National Oceanic and Atmospheric Administration (NOAA) Global Climate Observing System Surface Network (GSN) [3]. This data provides highly accurate daily data, but only at sites that have surface observing stations. For denser, more global coverage, we also use data from the Moderate Resolution Imaging Spectroradiometer (MODIS) instrument aboard NASA's Terra satellite. The satellite is in a polar orbit so that it scans the entire surface of the earth every day. The MODIS instrument measures spectral emissions at various wavelengths, and then post-processing uses these measurements to estimate ground cover. In this work we use two datasets: the daily snow cover maps [55] and the two-week vegetation averages [84]. Both of these sets of data including an estimate of the percentage of snow or vegetation ground cover at each point on earth, along with a quality score indicating the confidence in the estimate. Low confidence is caused primarily by cloud cover (which changes the spectral emissions and prevents accurate ground cover from being estimated), but also by technical problems with the satellite. As an example, Figure 1.5 shows raw satellite snow data from one particular day.

4.3.1.1 SNOW DATASET

The distribution of geo-tagged Flickr photos is highly non-uniform, with high peaks in population centers and tourist locations. Sampling uniformly at random from Flickr photos produces a dataset that mirrors this highly non-uniform distribution, biasing it towards cities and away from rural areas. Since our eventual goal is to reproduce continental-scale satellite maps, rural areas are very important. An alternative is biased sampling that attempts to select more uniformly over the globe, but has the disadvantage that it no longer reflects the distribution of Flickr photos. Other important considerations include how to find a variety of snowy and non-snowy images, including relatively difficult images that

may include wintry scenes with ice but not snow, and how to prevent highly-active Flickr users from disproportionately affecting the datasets.

We strike a compromise on these issues by combining together datasets sampled in different ways. We begin with a collection of about more than one hundred million Flickr photos geo-tagged within North America and collected using the public API (by repeatedly querying at different times and geo-spatial areas, similar to [60]). From this set, we considered only photos taken before January 1, 2009 (so that we could use later years for creating a separate test set), and selected: (1) all photos tagged *snow, snowfall, snowstorm,* or *snowy* in English and 10 other common languages (about 500,000 images); (2) all photos tagged *winter* in English and about 10 other languages (about 500,000 images); (3) a random sample of 500,000 images. This yielded about 1.4 million images after removing duplicates. We further sampled from this set in two ways. First, we selected up to 20 random photos from each user, or all photos if a user had less than 20 photos, giving about 258,000 images. Second, we sampled up to 100 random photos from each $0.1^\circ \times 0.1^\circ$ latitude-longitude bin of the earth (roughly 10km $\times$ 10km at the mid latitudes), yielding about 300,000 images. The combination of these two datasets has about 425,000 images after removing duplicates, creating a diverse and realistic selection of images. We partitioned this dataset into test and training sets on a per-user basis, so that all of any given user's photos are in one set or the other (to reduce the potential for duplicate images appearing in both training and test).

We then presented a subset of these images to humans and collected annotations for each image. We asked people to label the images into one of four categories: (1) contains obvious snow near the camera; (2) contains a trace amount of snow near the camera; (3) contains obvious snow but far away from the camera (e.g., on a mountain peak); and (4)

does not contain snow. For our application of reconstructing snowfall maps, we consider (1) and (2) to be positive classes and (3) and (4) to be negative, since snowfall in the distance does not give evidence of snow at the image's geo-tagged location. In total we labeled 10,000 images.

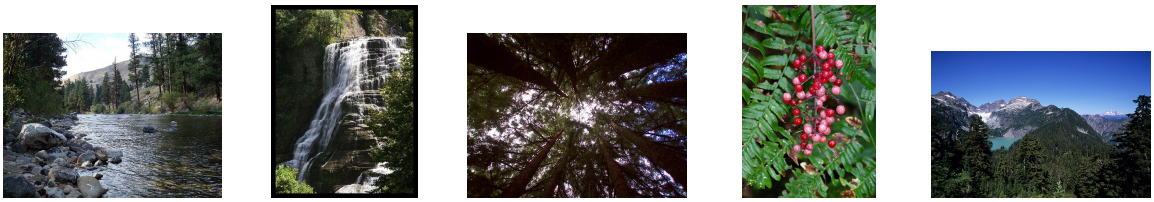
4.3.1.2 VEGETATION DATASET

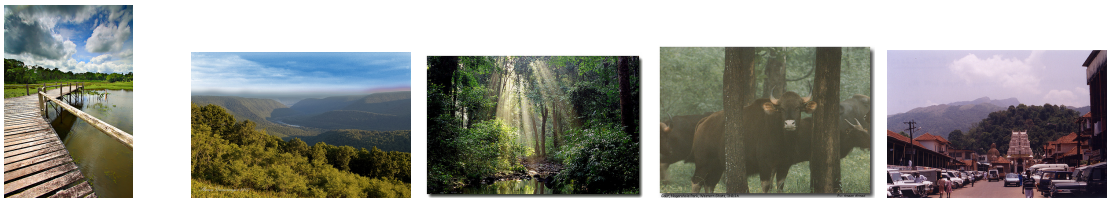
We build a data set with over 10,000 images. They are taken before 2009, and are composed of images with "forest" and "summer" tags and also a collection of random images without any tag preference. These images were hand-labeled with categories *"Outdoor Greenery, Outdoor non-Greenery, Indoor, and Other"*.

Finally, we build a positive set with images in the category *"Outdoor Greenery"* and a negative set with images in categories *"Outdoor non-Greenery"* and *"Indoor"* to learn an image classification model, we build a training set with 4,000 images and a testing set with 2000 images. In training and testing set, there are equal numbers of positive and negative samples. To show the diversity of our Flickr image dataset, in Figure 4.2 we present a random sample of images in our vegetation dataset labeled as positive and negative.

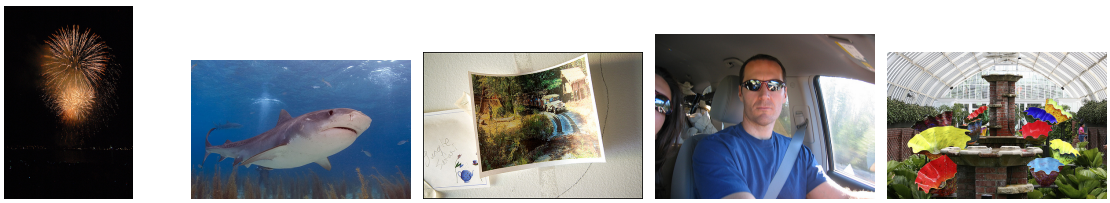
4.3.2 EXTRACTING SEMANTICS USING TAGS FROM INDIVIDUAL IMAGES

We consider two learning paradigms. The first is to produce a single exemplar for each bin in time and space consisting of the set of all tags used by all users. For each of these exemplars, the NASA and/or NOAA ground truth data gives a label (snow or non-snow). We then use standard machine learning algorithms like Support Vector Machines and decision trees to identify the most discriminative tags and tag combinations. In the second paradigm, our goal instead is to classify individual *photos* as containing snow or not, and





(a) Random positive images in vegetation dataset



(b) Random negative images in vegetation dataset

Figure 4.2: Random images from our hand-labeled dataset. Public sharing images vary in quality, contents, illumination and view angle. Negative images like winter trees without leaves, or indoor images capturing a photo of forest are more confusing.

then use these classifier outputs to compute the number of positive and non-positive photos in each bin.

4.3.3 EXTRACTING SEMANTICS USING VISUAL FEATURES FROM INDIVIDUAL IMAGES

Snow is a somewhat unique visual phenomenon, and we claim that detecting it in images is a unique recognition task. In some cases, snow can be detected by coarse scene recognition: ski slopes or snowy landscapes are distinctive scenes. But snow can appear in any kind of outdoor scene, and is thus like an object. However, unlike most objects that have some distinctive features, snow is simply a white, near-textureless material. (In fact, our informal observation is that humans detect snow not by recognizing its appearance, but by noticing that other expected features of a scene are occluded; in this sense, detecting snow is less about the features that are seen and more about the features that are *not* seen. We leave this as an observation to inspire future work.) We tested a variety of off-the-shelf visual features for classifying whether an image contains fallen snow. We used Support Vector Machines for classification, choosing kernels based on the feature type. Intuitively, color is a very important feature for detecting snow, and thus we focused on features that use color to some degree.

Similar to snow, vegetation has a signature color (green). The leaves of plants also have distinctive visual texture. So we employ SIFT features to analyze the local gradient distribution, and we also extract GIST feature to describe texture feature and global context. We used a wide variety of visual features, including those introduced in Chapter 2. Here we are describe the details of our visual features:

- **Color histograms:** We begin with perhaps the simplest of color features. We build

joint histograms in CIELAB space, with 4 bins on the lightness dimension and 14 bins along each of the two color dimensions, for a total of 784 bins. We experimented with other quantizations and found that this arrangement worked best. We encode the histogram as a 784 dimensional feature and use an SVM with a chi-squared distance (as in [142]).

- **Tiny images:** We subsample images to 16×16 pixels, giving 256 pixels per RGB color plane and yielding a 768 dimensional feature vector. Drastically reducing the image dimensions yields a feature that is less sensitive to exact alignment and more computationally feasible [133].
- **Spatial Moments:** Tiny images capture coarse color and spatial scene layout information, but much information is discarded during subsampling. As an alternative approach, we convert the image to LUV color space, divide it into 49 blocks using a 7×7 grid, and then compute the mean and variance of each block in each color channel. Intuitively, this is a low-resolution image and a very simple texture feature, respectively. We also compute maximum, minimum, and median value within each cell, so that the final feature vector has 735 dimensions.
- **Color Local Binary Pattern (LBP) with pyramid pooling:** LBP represents each 9×9 pixel neighborhood as an 8-bit binary number by thresholding the 8 outer pixels by the value at the center. We build 256-bin histograms over these LBP values, both on the grayscale image and on each RGB color channel Korayem *et al* [79]. We compute these histograms in each cell of a three-level spatial pyramid, with 1 bin at the lowest level, 4 bins in a 2×2 grid at the second level, and 16 bins in a 4×4 grid at the third level. This yields a $(1 + 4 + 16) \times 4 \times 256$ = 21504 dimensional feature vector for each image.

- **GIST:** We also apply GIST features, which capture coarse texture and scene layout by applying a Gabor filter bank followed by down-sampling Oliva and Torralba [107]. Our variant produces a 1536-dimensional vector and operates on color planes. Scaling images to have square aspect ratios before computing GIST improved classification results significantly [41].
- **Color SIFT histogram:** We extract dense SIFT feature on each of the RGB color plane, and concatenate them to build color SIFT feature. The dense SIFT feature is extracted from every 2 pixels by 2 pixels bin, with a step size of 5 pixels. In this way, we achieve representative key points and reasonable computation complex. From training data set, we build 2,000 dimensional centers of color SIFT features using K-means clustering. With these centers, a 2000 dimensional histogram is built from all the key points of each image. Using the SIFT histograms, a model is trained and tested with SVM using RBF kernel.

We also experimented with a number of other features, and found that they did not work well on snow detection; local features like SIFT and HOG in particular perform poorly, again because snow does not have distinctive local visual appearance.

4.3.3.1 DEEP LEARNING

We apply CNNs to detect snow and vegetation on an image level. We followed Oquab [108] et al. and started with a model pre-trained on the ImageNet dataset, and then we train our models using hand-labeled data sets.

4.3.4 COMBINING EVIDENCE TOGETHER ACROSS USERS

Using image classification techniques, we can predict the snow or vegetation for each image, however our goal is to estimate the presence or absence of a given ecological phe-

nomenon (like a species of plant or flower, or a meteorological feature like snow) on a given day and at a given place, using only the geo-tagged, time-stamped photos from Flickr. One way of viewing this problem is that every time a user takes a photo of a phenomenon of interest, they are casting a "vote" that the phenomenon actually occurred in a given geospatial region. We could simply look for tags indicating the presence of a feature – i.e. count the number of photos with the tag "snow" – but sources of noise and bias make this task challenging, including:

— *Sparse sampling:* The geospatial distribution of photos is highly non-uniform. A lack of photos of a phenomenon in a region does not necessarily mean that it was not there.

— *Observer bias:* Social media users are younger and wealthier than average, and most live in North America and Europe.

— *Incorrect, incomplete and misleading tags:* Photographers may use incorrect or ambiguous tags — e.g. the tag "snow" may refer to a snowy owl or interference on a TV screen.

— *Measurement errors:* Geo-tags and timestamps are often incorrect (e.g. because people forget to set their camera clocks).

The second component (probabilistic model) in our framework is designed to deal with these types of noise and bias.

A probablistic model. We introduce a simple probabilistic model and use it to derive a statistical test that can deal with some such sources of noise and bias. The test could be used for estimating the presence of any phenomenon of interest; without loss of generality we use the particular case of snow here, for ease of explanation. Any given photo either contains evidence of snow (event s) or does not contain evidence of snow (event $\bar{s}$). We

assume that a given photo taken at a time and place with snow has a fixed probability $P(s|snow)$ of containing evidence of snow; this probability is less than 1.0 because many photos are taken indoors, and outdoor photos might be composed in such a way that no snow is visible. We also assume that photos taken at a time and place without snow have some non-zero probability $P(s|\overline{snow})$ of containing evidence of snow; this incorporates various scenarios including incorrect timestamps or geo-tags and misleading visual evidence (e.g. man-made snow).

Let m be the number of snow photos (event s), and n be the number of non-snow photos (event $\bar{s}$) taken at a place and time of interest. Assuming that each photo is captured independently, we can use Bayes' Law to derive the probability that a given place has snow given its number of snow and non-snow photos,

$$\begin{aligned} P(snow|s^m, \bar{s}^n) &= \frac{P(s^m, \bar{s}^n|snow)P(snow)}{P(s^m, \bar{s}^n)} \\ &= \frac{\binom{m+n}{m}p^m(1-p)^n P(snow)}{P(s^m, \bar{s}^n)}, \end{aligned}$$

where we write $s^m, \bar{s}^n$ to denote m occurrences of event s and n occurrences of event $\bar{s}$, and where $p = P(s|snow)$ and $P(snow)$ is the prior probability of snow. A similar derivation gives the posterior probability that the bin does not contain snow,

$$P(\overline{snow}|s^m, \bar{s}^n) = \frac{\binom{m+n}{m}q^m(1-q)^n P(\overline{snow})}{P(s^m, \bar{s}^n)},$$

where $q = P(s|\overline{snow})$. Taking the ratio between these two posterior probabilities yields a likelihood ratio,

$$\frac{P(snow|s^m, \bar{s}^n)}{P(\overline{snow}|s^m, \bar{s}^n)} = \frac{P(snow)}{P(\overline{snow})}\left(\frac{p}{q}\right)^m\left(\frac{1-p}{1-q}\right)^n. \tag{4.1}$$

This ratio can be thought of as a measure of the confidence that a given time and place actually had snow, given photos from Flickr.

A simple way of classifying a photo into a positive event s or a negative event $\bar{s}$ is to use text tags or output of image classifier. In case of using tags, we identify a set $\mathcal{S}$ of tags related to a phenomenon of interest. Any photo tagged with at least one tag in $\mathcal{S}$ is declared to be a positive event s, and otherwise it is considered a negative event $\bar{s}$. For the snow detection task, we use the set $\mathcal{S}$={snow, snowy, snowing, snowstorm}, which we selected by hand.

The above derivation assumes that photos are taken independently of one another, which is generally not true in reality. One particular source of dependency is that photos from the same user are highly correlated with one another. To mitigate this problem, instead of counting m and n as numbers of *photos*, we instead let m be the number of *photographers* having at least one photo with evidence of snow, while n is the numbers of photographers who did not upload any photos with evidence of snow.

The probability parameters in the likelihood ratio of equation (4.1) can be directly estimated from training data and ground truth. For example, for the snow cover results presented in Section 4.4, the learned parameters are: $p = p(s|snow) = 17.12\%$, $q = p(s|\overline{snow}) = 0.14\%$. In other words, almost 1 of 5 people at a snowy place take a photo containing snow, whereas about 1 in 700 people take a photo containing evidence of snow at a non-snowy place.

4.4 EXPERIMENTS AND RESULTS

In this section, we show our experiments to evaluate our framework. We present the results for single image classification component as well as our probabilistic model. We

Table 4.1: Performance of different features for snow detection.

Feature	Kernel	Accuracy
Random Baseline	—	50.0%
Gist	RBF	73.7%
Color	χ^2	74.1%
Tiny	RBF	74.3%
Spatial Color Moments	RBF	76.2%
Spatial pyramid LBP	RBF	**77.0%**
All traditional features	linear	**80.5%**
CNN	-	**88.1%**

evaluate our methods using two cases: snow and vegetation.

4.4.1 SNOW

4.4.1.1 SINGLE IMAGE CLASSIFICATION

We used a variety of visual features for classifying whether an image contains fallen snow. We used Support Vector Machines for classification, choosing kernels based on the feature type. We also apply CNN and extract the features and classify the images.

We tested these approaches to detecting snow on our dataset of 10,000 hand-labeled images. We split this set into a training set of 8,000 images and a test set of 2,000 images, sampled to have an equal proportion of snow and non-snow images (so that the accuracy of a random baseline is 50%). Table 4.1 presents the results. We observe that all of the features perform significantly better than a random baseline. Gist, Color Histograms and Tiny Image all give very similar accuracies, within a half percentage point of 74%. Spatial Moments and LBP features perform slightly better at 76.2% and 77.0%. We also tested a combination of all features by learning a second-level linear SVM on the output of the five SVMs; this combination performed significantly better than any single traditional feature,

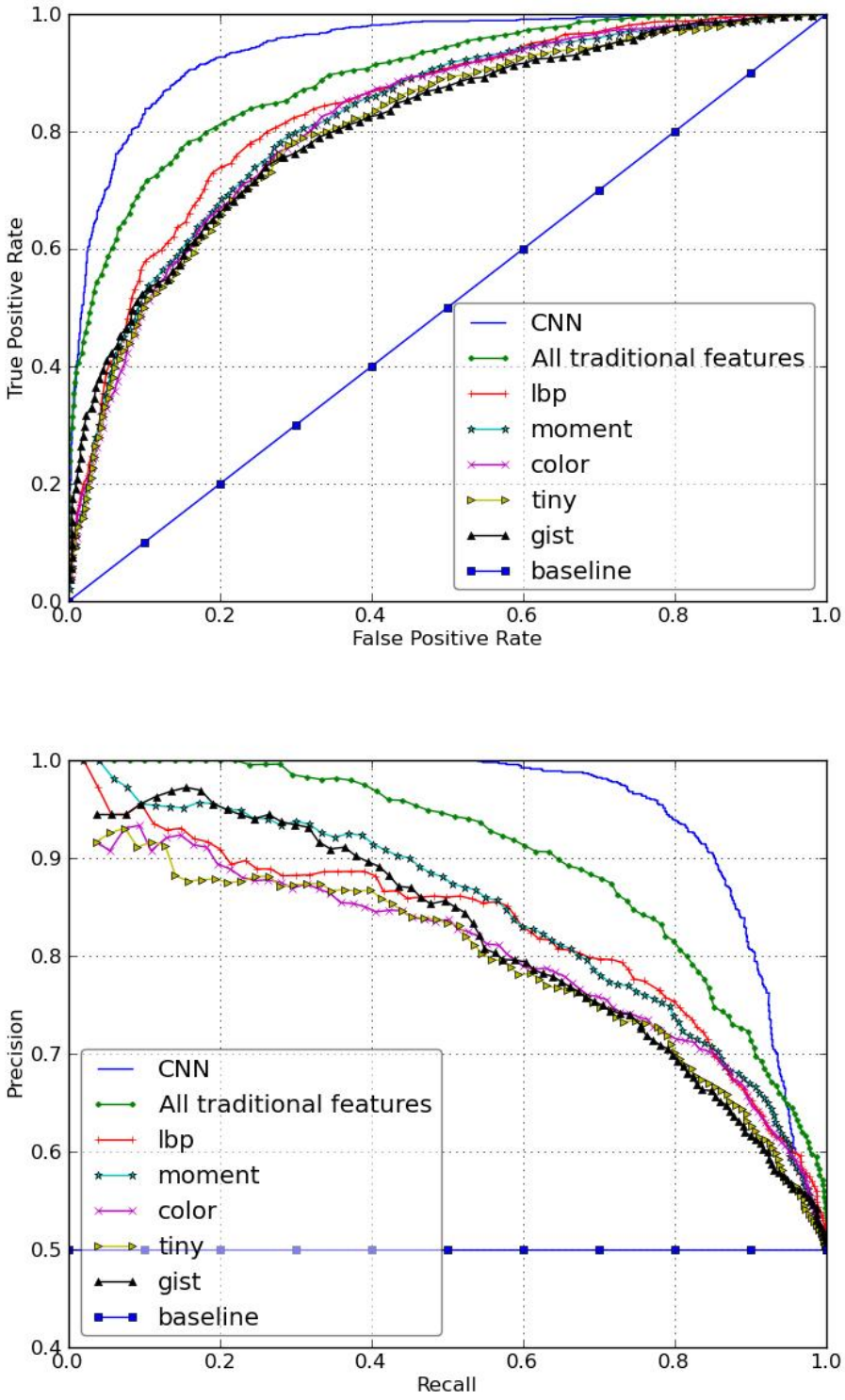


Figure 4.3: Snow classification results for different features and combinations, in terms of *(top):* ROC curves for the task of classifying snow vs. non-snow images; and *(bottom):* Precision-Recall curves for the task of retrieving snow images.

at 80.5%. The best performance we obtained was through using deep features which is 8% better than the combination of the traditional features.

Figure 4.3 shows classification performance in terms of an ROC curve, as well as a precision-recall curve in which the task is to retrieval photos containing snow. The precision-recall curve shows that at about 60% recall, precision is very near to 100%, while even at 85% recall, precision is close to 90%. This is a nice feature because in many applications, it may not be necessarily to correct classify all images, but instead to find some images that most likely contain a subject of interest.

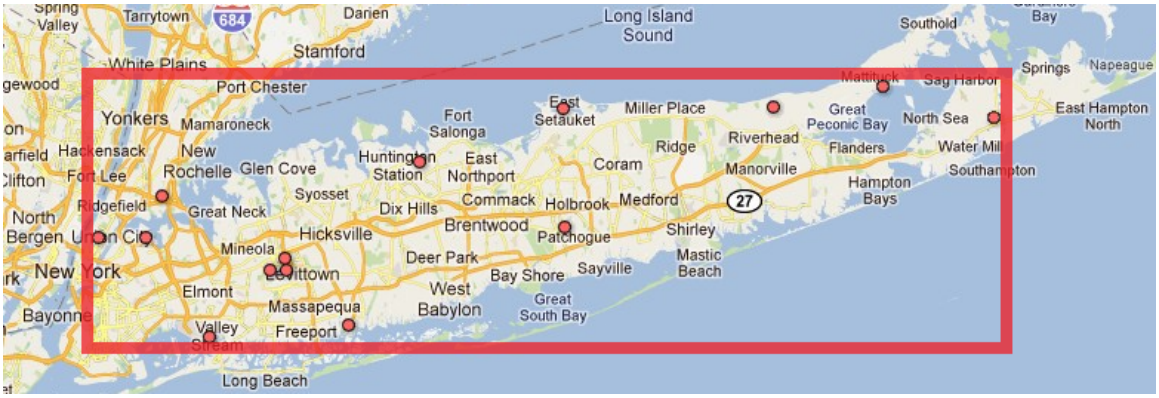
Our best performance using our traditional visual features is 80% accuracy. We also built CNN visual model for snow detection. We start training our network using ImageNet pre-trained model weights [112], and then we train our model using our shown training data. CNN achieves 88% accuracy which outperforms all other features by 8%. Therefore, we used CNN as our visual model for the final predictions. Similar to our visual model, we also built SVM using only tags as features and our text classifier achieves 87%.

We now turn to presenting experimental results for estimating the geo-temporal distributions of snow.

4.4.1.2 SNOW PREDICTION IN CITIES

We first test how well the Flickr data can predict snowfall at a local level, and in particular for cities in which high-quality surface-based snowfall observations exist and for which photo density is high.

We choose 4 U.S. metropolitan areas, New York City, Boston, Chicago and Philadelphia, and try to predict both daily snow presence. For each city, we define a corresponding geospatial bounding box and select the NOAA ground observation stations in that area.



	NYC	Chicago	Boston	Philadelphia
Mean active Flickr users / day	65.6	94.9	59.7	43.7
Approx. city area (km^2)	3,712	11,584	11,456	9,472
User density (avg users/unit area)	112.4	52.5	33.5	29.6
Mean daily snow (inches)	0.28	0.82	0.70	0.35
Snow days (snow>0 inches)	185	418	373	280
Number of obs. stations	14	20	41	26

Figure 4.4: *Top:* New York City geospatial bounding box used to select Flickr photos, and locations of NOAA observation stations. *Bottom:* Statistics about spatial area, photo density, and ground truth for each of the 4 cities.

For example, Figure 4.4 shows the stations and the bounding box for New York City. We calculate the ground truth daily snow quantity for a city as the average of the valid snowfall values from its stations. We call any day with a non-zero snowfall or snowcover to be a snow day, and any other day to be a non-snow day.

Figure 4.4 also presents some basic statistics for these 4 cities. All of our experiments involve 4 years (1461 days) of data from January 2007 through December 2010; we reserve the first two years for training and validation, and the second two years for testing.

Daily snow classification for 4 cities. Figure 4.5(a) presents ROC curves for this daily snow versus non-snow classification task on New York City. The figure compares the likelihood ratio confidence score from equation (4.1) to the baseline approaches (voting and

Table 4.2: Results for Confidence score model using tags and visual classifiers for our 4 cities.

City	baseline	tags	tag confidence	vision conf	tags conf and vision conf
NYC	85.00%	85.75 %	90.42 %	90.28 %	92.33 %
Chicago	72.80%	93.56 %	94.11 %	93.16 %	95.07 %
Boston	75.60%	90.54 %	89.17 %	85.20 %	91.23 %
Philly	80.50%	85.34%	89.19 %	85.08 %	89.19 %

percentage), using the tag set $\mathcal{S}$={snow, snowy, snowing, snowstorm}. The area under the ROC curve (AUC) statistics are 0.929, 0.905, and 0.903 for confidence, percentage, and voting, respectively, and the improvement of the confidence method is statistically significant with $p = 0.0713$ according to the statistical test of [138]. The confidence method also outperforms other methods for the other three cities (not shown due to space constraints). ROC curves for all 4 cities using the likelihood scores are shown in Figure 4.5(b). Chicago has the best performance and Philadelphia has the worst; a possible explanation is that Chicago has the most active Flickr users per day (94.9) while Philadelphia has the least (43.7).

We also tried training a classifier to learn these relationships automatically. For each day in each city, we produce a single binary feature vector indicating whether or not a given tag was used on that day. Also we tried to build classifiers trained based on our likelihood ratio computed based on tags or our visual model predictions. Table 4.2 shows the results for these classifiers. Best performance is obtained when we combine the confidence scores of tags and visual model based on CNN.

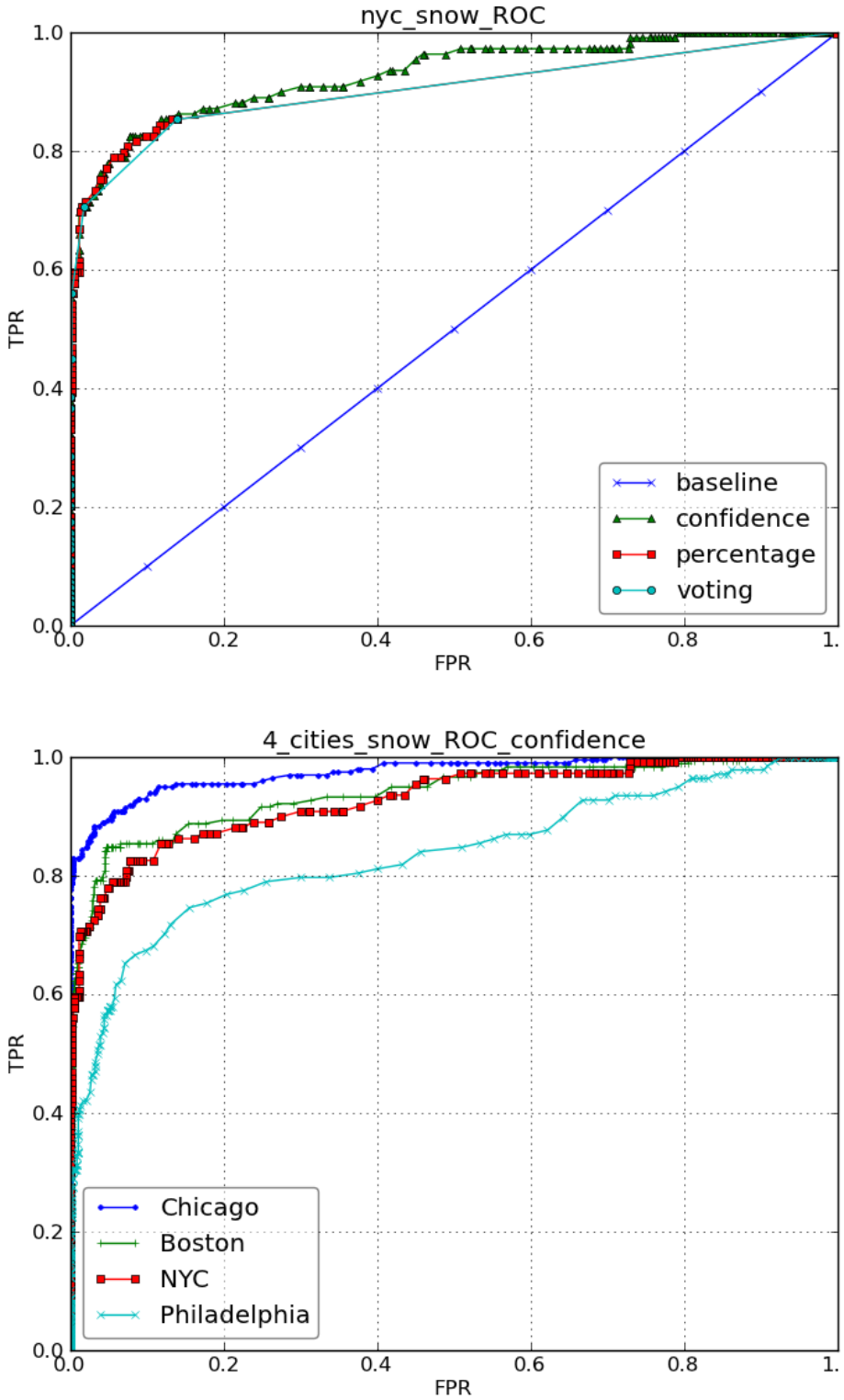


Figure 4.5: ROC curves for binary snow predictions. Top: ROC curves for New York City, comparing likelihood ratio confidence score to voting and percentage approaches, bottom: ROC curves for 4 cities using the likelihood scores

4.4.1.3 CONTINENTAL-SCALE SNOW PREDICTION

Here we ask whether phenomena can be monitored at a continental scale, a task for which existing data sources are less complete and accurate. We use the photo data and ground truth described earlier, although for these experiments, we restrict our dataset to North America (which we defined to be a rectangular region spanning from 10 degrees north, -130 degrees west to 70 degrees north, -50 degrees west). (We did this because Flickr is a dominant photo-sharing site in North America, while other regions have other popular sites — e.g. Fotolog in Latin America and Renren in China.)

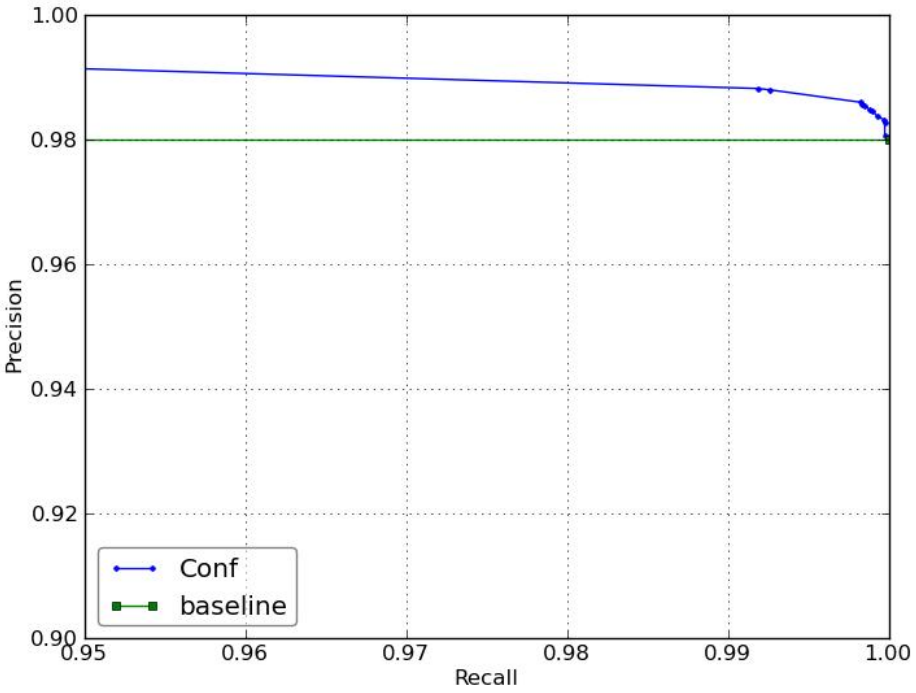
The spatial resolution of the NASA satellite ground truth datasets is 0.05 degrees latitude by 0.05 degrees longitude, or about $5 \times 5 km^2$ at the equator. (Note that the surface area of these bins is non-uniform because lines of longitude get closer together near the poles.) However, because the number of photos uploaded to Flickr on any particular day and at any given spatial location is relatively low, and because of imprecision in Flickr geo-tags, we produce estimates at a coarser resolution of 1 degree square, or roughly $100 \times 100 km^2$. To make the NASA maps comparable, we downsample them to this same resolution by averaging the high confidence observations within the coarser bin. We then threshold the confidence and snow cover percentages to annotate each bin with one of three ground truth labels:

— Snow bin, if confidence is above 90 and coverage above 80,

— Non-snow bin, if confidence is above 90 and coverage is 0,

— Unknown bin, otherwise.

Figure 4.6 shows the precision and recall curve of snow and non snow prediction in continental-scale. Here we limit our predictions for the bins which have photos, we do this by keeping the bins have ground truth and photos at the same time. We computed



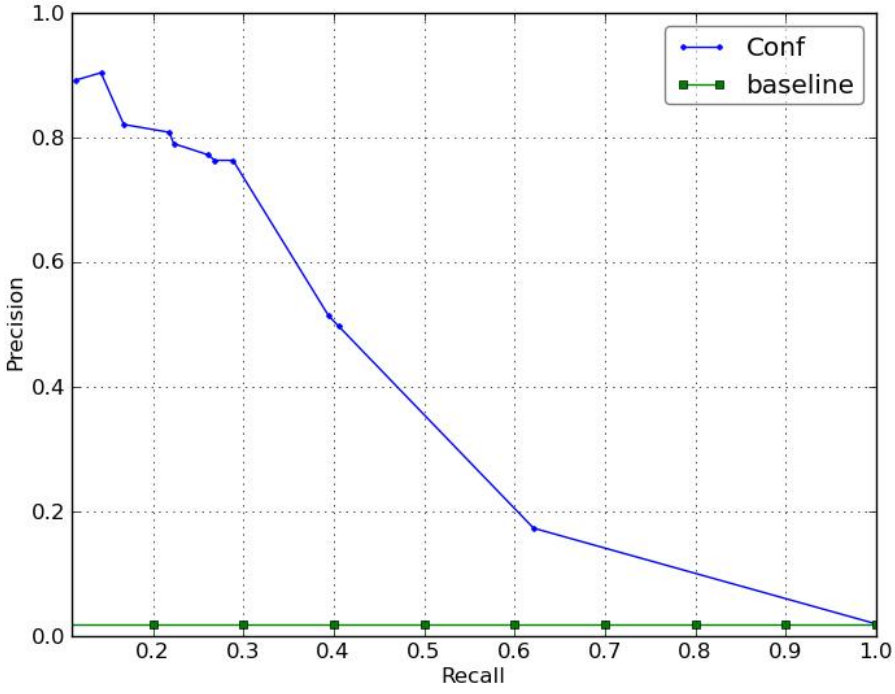


Figure 4.6: Precision and recall curve of snow and nonsnow prediction in continental scale.

Table 4.3: Results for our visual models for vegetation.

Visual feature	Accuracy
Random Baseline	50.0%
Color SIFT	78.1%
Color GIST	82.6%
SIFT and GIST	85.9%
CNN	88.0%

our confidence scores based on tags and image-classification, and then we trained a simple decision tree to learn the correct thresholds to make final prediction. We achieve almost 0.5% over the baseline (cutting the error rate by more than 20%), where the baseline in our case is the majority class classifier which predicts non-snow all the time.

4.4.2 VEGETATION

4.4.2.1 SINGLE IMAGE CLASSIFICATION

Using the method we describe in Section 4.3, we train and test the vision model on our hand-labeled data set. There are 4,000 images in the training set and 2,000 images in the testing set. In both training and testing set, the number of positive and negative images are the same. Here we present the results at an image classification level.

4.4.2.2 VEGETATION COVERAGE OVER TIME AND SPACE

We combine all the evidence over space and time in North America from 2007 to 2010. We compute the confidence score described in Section 4.3. The prior probability of a place being covered by vegetation at some time is 75.2%. For an image taken from a place covered by green vegetation at that time, the probability of this image being a green image is

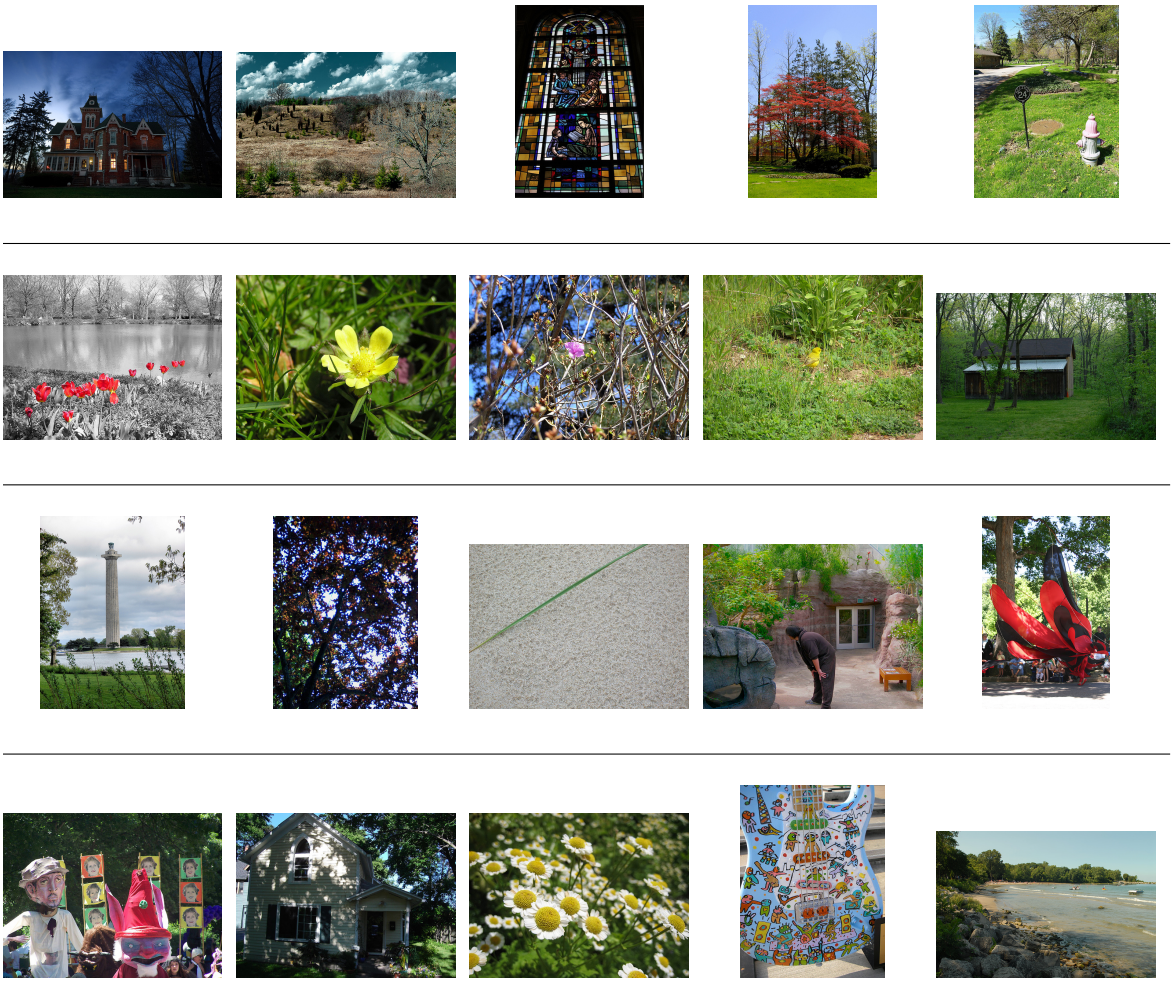


Figure 4.7: In vegetation detection over North America in 2009 and 2010, among all false positive bins, there are images that are predicted as greenery. These images are the reason these bins are predicted as green. Here are some random selected examples of the green images.

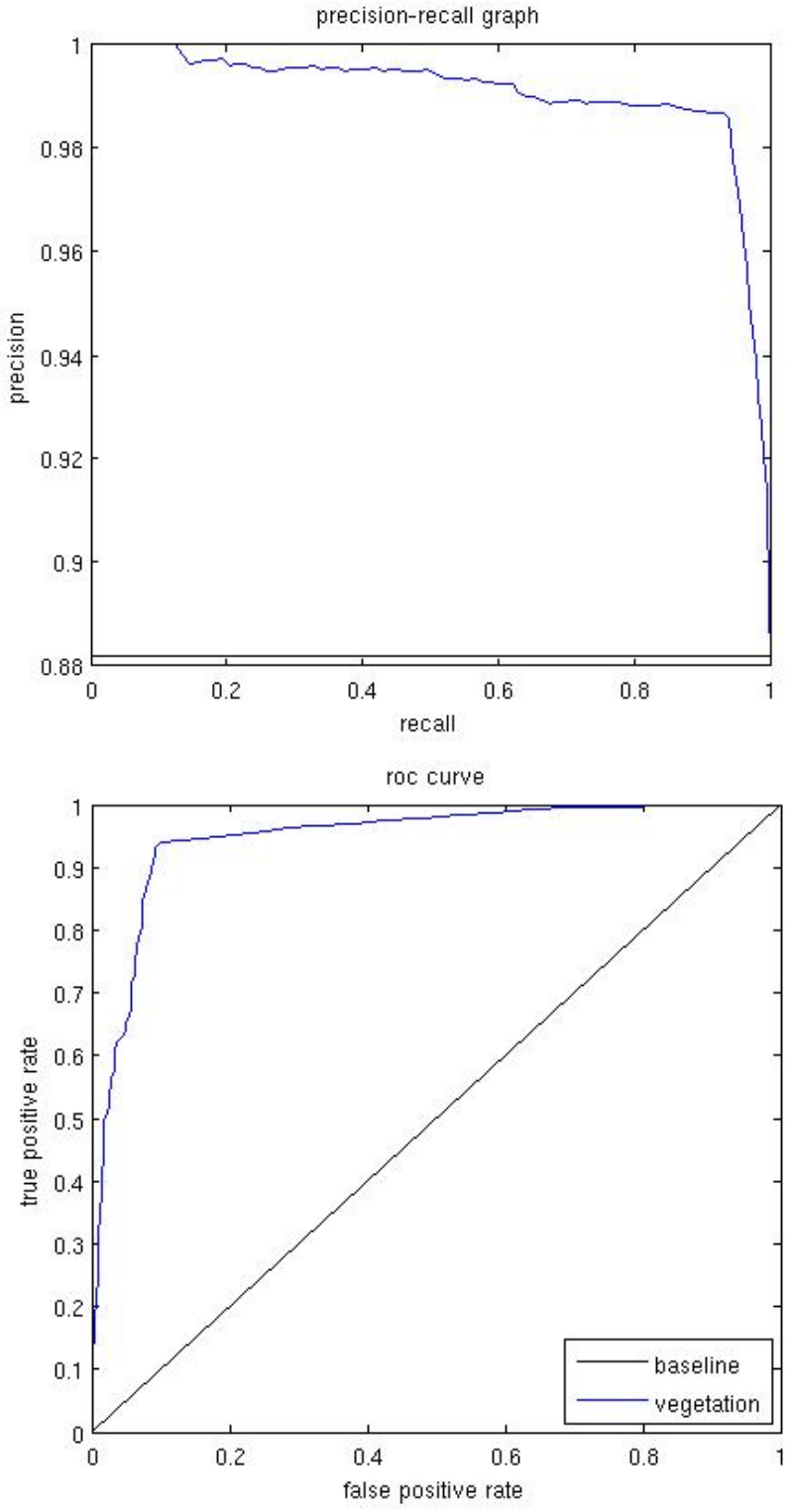


Figure 4.8: Top: Precision and recall curve of vegetation prediction in continental scale. Bottom: ROC curve of vegetation prediction in continental scale.

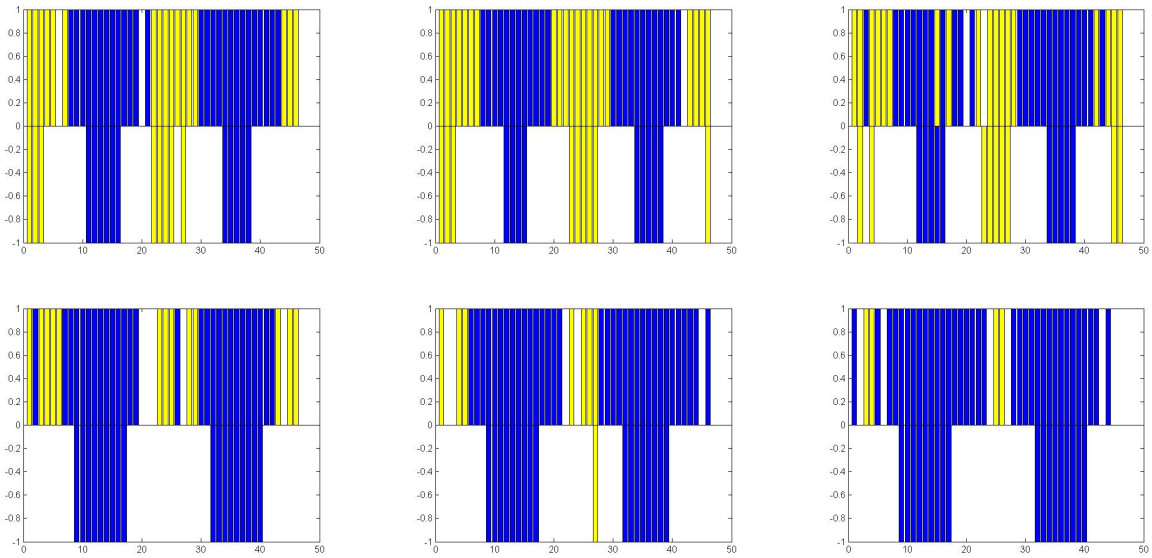


Figure 4.9: Yellow bars show non-greenery at that time. Blue bars represent greenery. Prediction results on top shows 6 random places comparing to satellite ground truth. The ground truth on the bottom tends to disappear when leaves are turning yellow or green.

27.18%. On the other hand, there is only 3.03% probability to see a green image in a place not covered by enough green vegetation at that time.

While the satellite has ground truth for 87,594 bins in North America, our method predicts 61,602 bins (70.3% in quantity). Moreover, about 20% of satellite ground truth is located in Northern Canada. On the other hand, our data is from users in social media. So our prediction focuses on more populated locations or places people like to visit such as natural scenery.

In North America, the overall accuracy of our method is 93.2% compared to the 86.6% majority baseline. The precision of green bins is 98.8% and the precision of non-green bins is 68.2%. Recall of green bins is 93.3% and recall of non-green bins is 92.5%. Figure 4.8 shows the precision and recall curve of greenery prediction in continental-scale.

Generally, all the false negative error is due to the sparseness of data. While not enough

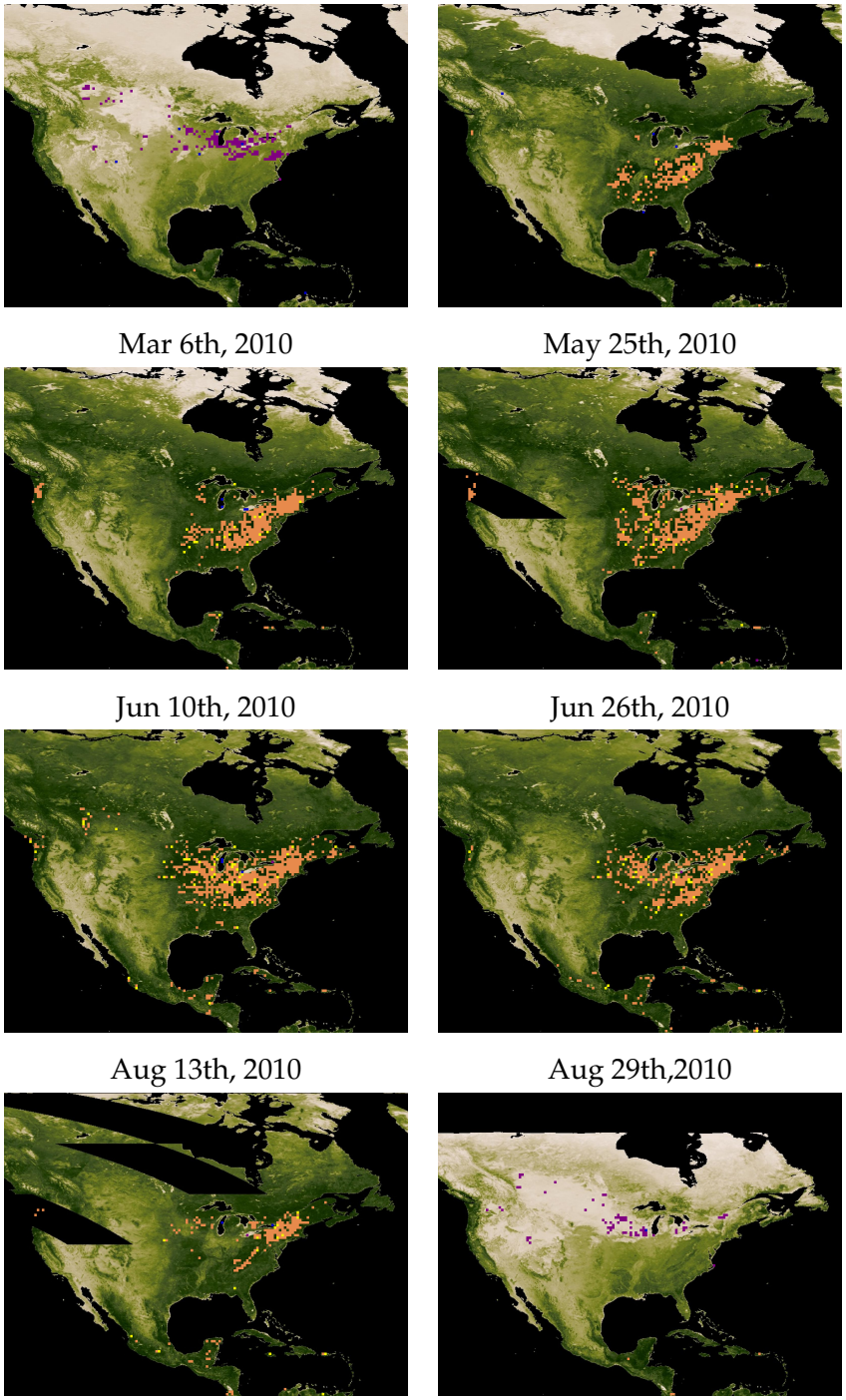


Figure 4.10: We use prediction results to recreate vegetation coverage maps for each 16-days period. There are 8 maps picked in 2010. The dates under each map are the starting date of each 16-days period. Orange bins are true positive; yellow bins are false negative; blue bins are false positive; and purple bins are true negative.

images are collected at certain location during some time, there is either no green image found or green images are too few compare to the quantity of non-green images. On the other hand, false positive errors are rare (less than 1%) and complex. We found most images in the false positive bins are actually green vegetation images. In Figure 4.7, we show some examples of images in false positive bins. Figure 4.9 shows vegetation coverage of 6 places over 2009 and 2010 and sample maps are presented in Figure 4.10. These maps are visualizations of the performance in North America.

4.5 SUMMARY

In this chapter we presented a system for mining photo-sharing website to study ecological phenomena. We proposed using the massive collections of user generated photos as a source of observational evidence about the world by automatically recognizing specific types of scenes and objects in large-scale social image collections.

Our framework consists of two components: image classification and probabilistic models (a Bayesian likelihood ratio). Our image classification component utilizes deep learning to classify individual images, and then a confidence score model to combine the evidence from multiple users and images over space and time to deal with noisy and biased sources existing in social media.

We study two different phenomena snow and vegetation coverage. Given the set of publicly-available Flickr photos (geo-tagged and time-stamped) at that place and time, we decide whether there was snow on the ground at a particular place and on a particular day. Also, we estimate vegetation coverage.

Our experiments suggest that mining from photo sharing websites could be a potential source of observational data for ecological and other scientific research. Our work is a

step towards that, however, there a lot of additional steps are needed to build a strong connection to real ecology applications. Our results show the precision can be quite high in spite of relatively low recall due to the sparsity of photos on social media.

CHAPTER 5

IMAGE CLASSIFICATION BASED SYSTEMS FOR PRIVACY APPLICATIONS

5.1 INTRODUCTION

In this chapter, we show systems for detecting life-logging images that are taken in private places or that contain sensitive objects. To detect sensitive locations, we develop methods to predict which room of a pretrained environment a life-logging photo is taken in. As a proof of concept, we also train classifiers to predict the presence of a computer screen as an example of a sensitive object. We discuss our methods in the context of user privacy, but these techniques are tools for photo organization and have applications outside of censoring images. Life-logging images are captured automatically at regular intervals and often have random content taken at unusual positions and scales. This diversity proves to increase the difficulty of the problem compared to canonically composed photographs.

Cameras are now ubiquitous, especially with the new generation of wearable devices (e.g., Google Glass, Autographer, Narrative Clip) (Figure 5.1). These wearable devices (Lifelogging devices) allow users to capture photos continuously (e.g., every 30 seconds on the Narrative Clip), recording the everyday moments in a user's environment.

Since lifelogging devices and applications record and may share images from daily life with their social networks, these devices raise serious privacy concerns [10]. Users of

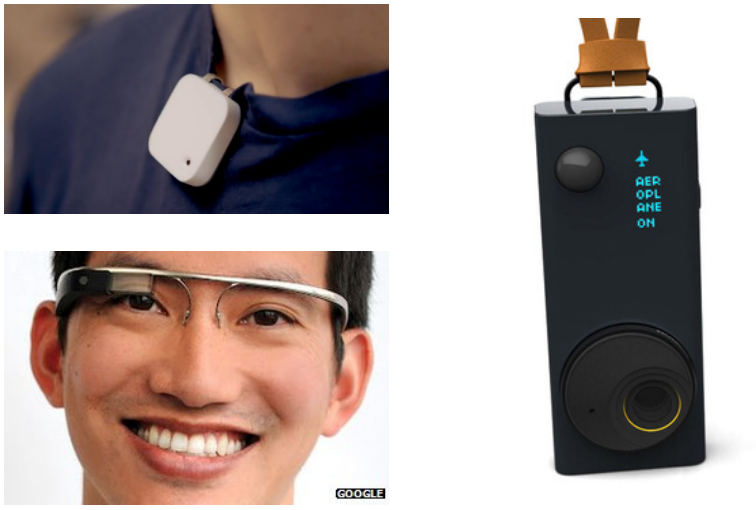


Figure 5.1: Wearable camera devices. *Clockwise from top left:* Narrative Clip takes photos every 30 seconds; Autographer has a wide-angle camera and various sensors; Google Glass features a camera, heads-up display, and wireless connectivity. (Photos by Narrative, Gizmodo, and Google.)

these devices needs solutions to help keep their private images safe and prevent sharing of sensitive images.

Here, we take first steps towards automatic detection and blocking of potentiality sensitive images. Detection of sensitive images is a very difficult problem, involving detecting and reasoning about image content, user activity, environmental context, social norms, etc. As a beginning, we develop two systems based on image classification and machine learning for screening images:

- PlaceAvoider [128] analyzes images to determine where they were taken, and to filter out images from places like bedrooms and bathrooms.
- ObjectAvoider [80] filters images based on their content, looking for objects that may signal privacy violations (e.g., computer monitors).

The rest of this chapter is organized as follows: Section 5.2 describes related work, then Section 5.3 and Section 5.4 describe our proposed systems and their evaluation. We conclude this chapter by a summary in Section 5.5.

5.2 RELATED WORK

5.2.1 LIFELOGGING ISSUES AND PRIVACY

Allen [10] and Cheng et al. [30] show that there are many legal issues related to lifelogging, many of which are privacy related. The user study in [75] validate their conclusions and show that users want control over the data that is collected and stored. While Hoyle et al. explore privacy issues for lifeloggers [65], Denning et al. consider the issues of bystanders of the lifeloggers [40]. These works show the needs for systems like PlaceAvoider and ScreenAvoider. Caine [23] shows that mistakes by users can lead to sharing the information with an unintended group. This type of problem is addressed by the proposed systems. The PlaceRaider system is a smartphone based attack that shows how opportunistically-collected images can be used by an adversary to reconstruct 3D models of their personal places. It shows the need for controls on the use of cameras on smartphones [129].

Chaudhari et al. [28] present a protocol for detecting and obscuring faces in a video stream. Many current lifelogging devices (e.g. Memoto [100] and Autographer [13]) and smartphone lifelogging apps provide advanced collection capabilities but have not considered many privacy concerns. These works motivate the necessity of the proposed systems to help users managing their life logging images.

5.2.2 IMAGE DEFENSES, CLASSIFICATION AND LOCALIZATION

There have been very few systems similar to our proposed system that try to control the collection of images. Truong et al. [134] demonstrate third-party system to detect and disable cameras via a directed pulsing light. This requires specialized infrastructure to be installed in each sensitive space. The PlaceAvoider proposed system can be integrated

within the camera to apply the same functionality.

Jana et al. present the DARKLY system [68] to add a privacy-protection layer to systems where untrusted applications can access camera resources. DARKLY use OpenCV within device middleware to control the amount of image content available to applications.

CrowdSense@Place [31] uses computer vision techniques and processing of recorded audio to classify location into one of seven general categories (e.g., home, workplace, shops). That is different from PlaceAvoider which try to apply fine indoor localization.

Most localization work in computer vision is for robotics applications [119, 120, 135]. Also, recent work studies geo-location in social images (consumer images). Most of this work studies highly photographed outdoor landmarks images. Other work use million of images to train models to recognize landmarks [48, 93, 94, 125]. Quattoni and Torralba classify images based on type of scene (e.g., indoors vs. outdoors) [111].

Most localization and positioning approaches require external infrastructure (e.g., audio) or dense of cooperating devices [61,118]. A comprehensive survey of these approaches is presented by Hightower [61].

5.3 PLACEAVOIDER

As a first step toward maintaining privacy in life-logging devices, we present PlaceAvoider, a system designed to help owners of first-person cameras 'blacklist' sensitive spaces (such as bathrooms and bedrooms). PlaceAvoider recognizes images taken in these spaces to flag them for review before making them available for applications or sharing on social networks. We first ask users to take images for sensitive spaces (e.g., bathrooms, bedrooms, offices), to build visual models for the sensitive space. PlaceAvoider uses both fine-grained image features (e.g., specific objects) and coarse-grained, scene-level features (like colors

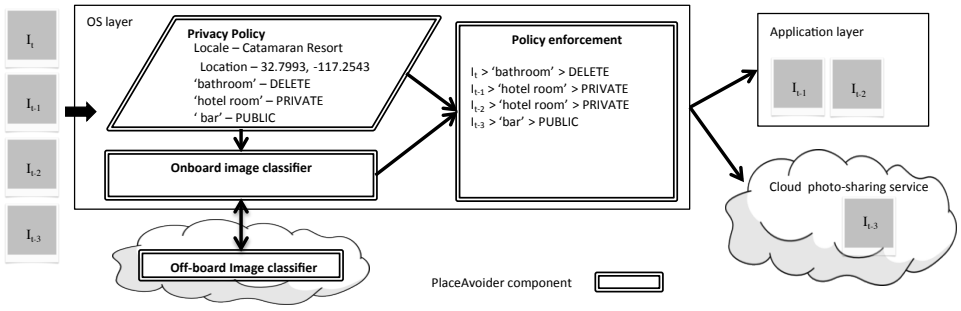


Figure 5.2: An abstract depiction of PlaceAvoider enforcing a fine-grained camera privacy policy. Our model leverages cloud computation to perform compute-intensive tasks. Cloud-based implementations of PlaceAvoider could also enforce privacy preferences for photo sharing sites.

and texture). PlaceAvoider can operate at the system level to provide a warning before photos are delivered to applications.

5.3.1 SYSTEM MODEL

We identify sensitive images by analyzing image content in conjunction with contextual information such as GPS location and time. We consider fine-grained image control based on physical space for the images. Our system can prevent applications from access sensitive images till users review them or it can tag them to be used by trusted applications.

Our system (Figure 5.2) consists of three components: a *privacy policy* to indicate private spaces, an *image classifier* to flag sensitive images, and a *policy enforcement mechanism* to determine how sensitive images are handled by the receiving applications.

- ***Privacy policy.*** We use a policy as a set of blacklisted spaces. Each space in the policy has an identifier (e.g, bathroom), geospatial location (e.g., latitude and longitude), visual model (constructed from enrollment images) and action to be taken with PlaceAvoider.
- ***Image classifier.*** The image classifier generates a visual model for enrolled spaces. It needs to deal with large amounts of noise including motion blur, poor composition,

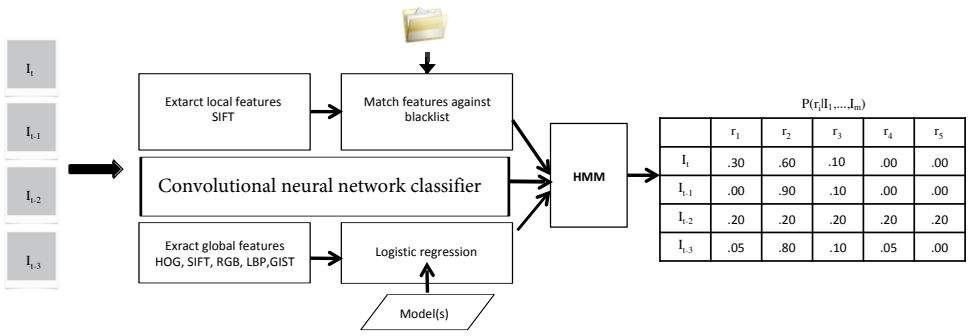


Figure 5.3: The PlaceAvoider classifier works on streams of images extracting local and global features. Single image classification feeds the HMM which outputs room labels and marginal distributions.

and occlusions. The classifier processes individual images, or jointly processes image sequences (image streams) to improve accuracy.

- ***Policy enforcement.*** We assume two mechanisms for policy enforcement. First is to block sensitive photos from applications, so that user can review these photos before they are delivered to the application. The second is to allow applications to use meta-data generated by the image classifier.

5.3.2 IMAGE CLASSIFICATION

The major component of PlaceAvoider is the image classifier (Figure 5.3). We assume that GPS provides a coarse position, so our goal here is to classify image content amongst a relatively small number of possible rooms within a known structure. While there is much work on scene and place recognition [94,142], we are not aware of work that has considered fine-grained indoor localization in images from first-person devices.

We first consider how to classify single images, using two complementary recognition techniques. Beside these two approaches, we also use a deep learning as a third type of classifier. We then show how to improve results by jointly classifying image sequences, taking advantage of temporal constraints on human motion.

5.3.2.1 CLASSIFYING INDIVIDUAL IMAGES

We apply three different approaches for classifying images. The first ('local classifier') is based on local invariant features (e.g. SIFT or SURF), where first interesting image points (like corners) are detected and represented in vectors that are insensitive to image transformations. The second approach ('global classifier') relies on global, scene-level image features, capturing broad color and texture patterns. The third approach is a CNN which has been proven to learn a good presentation of the image.

These approaches complement each other: local features work better in case of images having distinctive objects while global features and CNN try to model the overall structure of a room.

More formally, we assume that we have a small set $\mathcal{R} = \{r_1, ..., r_n\}$ of possible physical spaces (kitchen, bathroom, living room, etc.), and for each room r_i we have a set $\mathcal{I}_i$ of training images. Given a new image I, our goal is to assign it one of the labels in $\mathcal{R}$.

Local features. We represent each enrolled physical space as set of distinctive local feature descriptors which are invariant to variations such as scale and illumination. We use SIFT (Scale Invariant Feature Transform) [97] to generate these descriptors. SIFT works by finding interest points in images (e.g. corners), then computes the distribution of gradient orientations within small neighborhoods of each corner at different scales. The final output is a 128-dimensional invariant descriptor vector for each interest point. We generate a model for each room $r_i \in \mathcal{R}$ by extracting SIFT features of the training images which are taken in this room. We generate a set M_i for each room r_i using all feature lists of this room. Given a new image (test image) I, we need to find the best match using our models M_i. The simplest way for doing that is by counting how many points match for each room. Intuitively, this yields poor results as many common image features will exist in

different rooms. So we consider matching points based only on distinctive features using the following scoring function S that computes a similarity between a test image I and a given set of SIFT features M_i corresponding to the model of room r_i,

$$S(I, r_i) = \sum_{s \in I} \mathbb{1}\left(\frac{\min_{s' \in M_i} ||s - s'||}{\min_{s' \in M_{-i}} ||s - s'||} < \tau \right), \tag{5.1}$$

where M_{-i} is the set of features in all rooms except r_i, i.e. $M_{-i} = \cup_{r_j \in \mathcal{R} - \{r_i\}} M_j$, $\mathbb{1}(\cdot)$ is an indicator function that is 1 if its parameter is true and 0 otherwise, $|| \cdot ||$ denotes the L2 (Euclidean) vector norm, and τ is a threshold. This approach counts only distinctive features within each room and ignores common visual features across rooms.

Global features. Unfortunately, many first-person images do not have many distinctive features (e.g., blurry photos, photos of walls, etc.), causing local feature matching to fail since there are few features to match. We thus also use global, scene-level features that try to learn the general properties of a room, such as its color and texture patterns. These features can give meaningful hypotheses even for blurry and otherwise relatively featureless images. Instead of predefining a single global feature type, we instead compute a variety of features of different types and with different trade-offs as described in Chapter 2, and let the machine learning algorithm decide which of them are valuable for a given classification task. In particular, we use:

1. *RGB color histogram*, a simple 256-bin histogram of intensities over each of the three RGB color channels, which yields a 768-dimensional feature vector. This is a very simple feature that simply measures the overall color distribution of an image.
2. *Color-informed Local Binary Pattern (LBP)*, which converts each 3×3 pixel neighborhood into an 8-bit binary number by thresholding the 8 outer pixels by the value at

the center. We build a 256-bin histogram over these LBP values, both on the grayscale image and on each RGB channel, to produce a 1024-dimensional vector [79]. This feature produces a simple representation of an image's overall texture patterns.

3. *GIST,* which captures the coarse texture and layout of a scene by applying a Gabor filter bank and spatially down-sampling the resulting responses [41,107]. Our variant produces a 1536-dimensional feature vector.
4. *Bags of SIFT,* which vector-quantize SIFT features from the image into one of 2000 "visual words" (selected by running k-means on a training dataset). Each image is represented as a single 2000-dimensional histogram over this visual vocabulary [94, 142]. This feature characterizes an image in terms of its most distinctive points (e.g., corners).
5. *Dense bags of SIFT* are similar, except that they are extracted on a dense grid instead of at corner points and the SIFT features are extracted on each HSV color plane and then combined into 384-dimensional descriptors. We encode weak spatial configuration information by computing histograms (with a 300-word vocabulary) within coarse buckets at three spatial resolutions (1×1, 2×2, and 4×4 grid, for a total of $1 + 4 + 16 = 21$ histograms) yielding a 300×21 = 6,300-dimensional vector [142]. This feature characterizes an image in terms of both the presence and spatial location of distinctive points in the image.
6. *Bags of HOG* computes Histograms of Oriented Gradients (HOG) [37] at each position of a dense grid, vector-quantizes into a vocabulary of 300 words, and computes histograms at the same spatial resolutions as with dense SIFT, yielding a 6,300-dimensional vector. HOG features capture the orientation distribution of gradients in local neighborhoods across the image.

Once we extract features from labeled enrollment images, we learn classifiers using the LibLinear L2-regularized logistic regression technique [44].

Convolutional Neural Networks (CNN): The third type of classifier we used to classify a single image is a Convolutional Neural Network. To train our model, we started by a model pre-trained on the Places, a large scale dataset of 2,5 millions of images [144]. The model structure is very similar to the ImageNet model [108]. We then used our life logging training data to fine-tune the model parameters.

5.3.2.2 CLASSIFYING PHOTO STREAMS

The first-person camera devices that we consider here often take pictures at regular intervals, producing temporally ordered streams of photos. These sequences provide valuable contextual information because of constraints on human motion: if image I_i is taken in a given room, it is likely that I_{i+1} is also taken in that room. We thus developed an approach to jointly label sequences of photos in order to use temporal features as (weak) evidence in the classification. We use a probabilistic framework to combine this evidence in a principled way. By Bayes' law, the probability of a given image sequence having a given label sequence is,

$$P(l_1, ..., l_m | I_1, ..., I_m) \propto P(I_1, ..., I_m | l_1, ..., l_m) P(l_1, ..., l_m), \tag{5.2}$$

where the denominator of Bayes' law is ignored because the sequence is fixed (given to us by the camera). We make the following assumptions:

- The visual appearance of an image is conditionally independent from the appearance of other images given its room label, and
- The prior distribution over room label depends only on the label of the preceding image (the Markov assumption).

We can rewrite the probability in Equation 5.2 as,

$$P(l_1...l_m|I_1...I_m) \propto P(l_1)\prod_{i=2}^{m} P(l_i|l_{i-1})\prod_{i=1}^{m} P(I_i|l_i). \tag{5.3}$$

The first factor $P(l_1)$ represents the prior probability of the first room label. We ignore it because we assume here that is a uniform distribution. The second factor models the probability of a given sequence of room labels. This factor should capture the fact that humans are much more likely to stay in a room for several frames than to jump randomly from one room to the next. To model this fact, we use a very simple prior model,

$$P(l_i|l_{i-1}) = \begin{cases} \alpha, & \text{if } l_i \neq l_{i-1}, \\ 1-(n-1)\alpha, & \text{otherwise,} \end{cases}$$

where n is the number of classes (rooms) and α is a small constant (we use 0.01). Intuitively, this means that transitions from one room to another have much lower probability than staying in the same room. This prior model could be improved by considering contextual information about a place — e.g. it may be impossible to travel from the kitchen to the bedroom without passing through the living room first — but we do not consider that possibility in this paper.

The third factor in Equation (5.3) models the likelihood that a given image was taken in a given room. Intuitively, these likelihoods are produced by the local, global and CNN classifiers described above but we need to "convert" their outputs into probabilities. Again from Bayes' law,

$$P(I_i|l_i) = \frac{P(l_i|I_i)P(I_i)}{P(l_i)}.$$

We again ignore $P(I_i)$ (since I_i is observed and hence constant) and assume that the prior over rooms $P(l_i)$ is a uniform distribution, so it is sufficient to model $P(l_i|I_i)$. For the global classifiers, we use LibLinear's routines for producing a probability distribution $P_G(l_i|I_i)$ from the output of a multi-class classifier based on the relative distances to the class-separating hyperplanes [44] and we use the output of the soft-max layer for CNN classifier to generate a probability distribution $P_C(l_i|I_i)$. The output of local features is a matching score, and thus we introduce a simple probabilistic model to convert this score to probability. Equation (5.1) defined a score $S(I, r_i)$ between a given image I and a room r_i, in particular counting the number of distinctive image features in r_i that match I. This matching process is, of course, not perfect; the score will occasionally count a feature point as matching a room when it really does not. Suppose that the probability that any given feature match is correct is β, and is independent of the other features in the image. Now the probability that an image was taken in a room according to the local feature scores follows a binomial distribution,

$$P_L(l_i|I_i) \quad \propto \quad \binom{N}{S(I,l_i)} \beta^{S(I,l_i)}(1-\beta)^{N-S(I,l_i)}$$

where N is the total number of matches across all classes,

$$N = \sum_{r_i \in \mathcal{R}} S(I, r_i).$$

We set $\beta = 0.9$, the system is not very sensitive to this parameter unless it is set close to 0.5 (implying that correct matches are no more likely than chance) or to 1 (indicating that matching is perfect).

To produce the final probability $P(I_i|l_i)$, we multiply together $P_L(I_i|l_i)$, $P_G(I_i|l_i)$, and $P_C(I_i|l_i)$, treating local, global and CNN features as if they were independent evidence.

The model in Equation (5.3) is a Hidden Markov Model (HMM), and fast linear-time algorithms exist to perform inference. In this paper we use the HMM to perform two different types of inference, depending on the application (as described in Section 5.3.3). We may wish to find the most likely room label l_i^* for each image I_i given all evidence from the entire image sequence,

$$l_1^*, ..., l_m^* = \arg\max_{l_1,...,l_m} P(l_1, ..., l_m^*|I_1, ..., I_m),$$

which can be solved efficiently using the Viterbi algorithm [47]. In other applications, we may wish to compute the marginal distribution $P(l_i|I_1, ..., I_m)$ — i.e., the probability that a given image has a given label, based on all evidence from the entire image sequence — which can be found using the forward-backward algorithm [77]. The latter approach gives a measure of confidence; a peaky marginal distribution indicates that the classifiers and HMM are confident, while a flat distribution reflects greater uncertainty.

5.3.3 EVALUATION

To evaluate PlaceAvoider we performed several experiments to measure the performance of the system. Here we describe our first-person image datasets and then evaluate the performance of local, global and CNN classifiers on single images. Then, we evaluate the proposed stream classifier and reporting the computational performance.

Table 5.1: Summary of our datasets (enrollment photos). All datasets have five rooms (classes). Majority-class baselines are shown. For House 3, three rounds were taken with an HTC Amaze phone, one with a digital SLR camera, and one with a Samsung GT-S5360L phone.

Dataset	Device	resolution	# of images	# of rounds	Mean images/room	accuracy
House 1	iPhone 4S	8MP	184	3	61	22.8%
House 2	iPhone 5	8MP	248	3	83	29.9%
House 3	(see caption)	3-6MP	255	5	85	30.2%
Workplace 1	Motorola EVO	5MP	733	3	244	24.4%
Workplace 2	HTC Amaze	6MP	323	5	108	25.4%

5.3.3.1 EVALUATION DATASETS

We collected five new datasets of indoor spaces by four different users. Users collected the data independently. For each one of our datasets, we collected enrollment (training) images for each room. We tried to take a sufficient number of images for each room at different times to capture temporal variation. For each room we took three to five rounds of images. The total number of training images per space varied from 37 to 147, depending on the size of the room and the user.

For testing, we collected stream datasets where the users wore smart-phones on a lanyard around their neck. These smart phones are simulators of first-person cameras devices. These smart phones ran an app that took photos at a fixed interval (approximately every three seconds). Each collection duration ranged from about 15 minutes to one hour.

Here is the detailed descriptions of the datasets: We have three home and two workplace environments. Each dataset has five classes (rooms), Table 5.1 and Table 5.2 present detailed statistics on the datasets.

- ***House 1,*** a well-organized family home with three bedrooms, bathroom, and study;
- ***House 2,*** a sparsely-furnished single person's home, with garage, bedroom, office,

Table 5.2: Summary of our datasets (test photo streams). All datasets have five rooms (classes). Majority-class baselines are shown.

Dataset	Device	resolution	# of images	accuracy
House 1	iPhone 4S	8MP	323	29.8%
House 2	iPhone 5	8MP	629	31.0%
House 3	HTC Amaze	6MP	464	20.9%
Workplace 1	HTC Amaze	6MP	511	32.1%
Workplace 2	HTC Amaze	6MP	457	28.9%

bathroom, and living room;

- ***House 3,*** a somewhat more cluttered family home with two bedrooms, a living room, kitchen, and garage;
- ***Workplace 1,*** a modern university building with common area, conference room, bathroom, lab, and kitchen;
- ***Workplace 2,*** an older university building with a common area, conference room, bathroom, lab, and office.

5.3.3.2 SINGLE IMAGE CLASSIFICATION

Local features. We evaluate the local classifier using cross-validation approach to test the effect of parameters. If the dataset has r rounds of enrollment photos, we train r classifiers. For each classifier we used $r - 1$ rounds as the training images and the other round as the test one, and then we average the accuracies.

Table 5.3 shows the results for 5-way classification for our datasets on the original size images, while Table 5.4 shows the effect of downsampling the size of the images. Decreasing image resolution does not harm the classifier. It works almost the same as the raw resolution. Accuracy for original size images varies between 98.4% accuracy for House 1

Table 5.3: Local feature classifier trained and tested on enrollment images (Native-sized images) using cross-validation.

Dataset	Baseline accuracy	Classification accuracy	Mean # of features	# of images with ties	# of images with 5-way tie
House 1	22.8%	98.4%	297	2	0
House 2	29.9%	76.2%	209	27	8
House 3	30.2%	95.7%	59	12	5
Workplace 1	24.4%	84.0%	33	115	45
Workplace 2	25.4%	92.9%	104	15	6
Average	26.5%	89.4%	—	—	—

Table 5.4: Local feature classifier trained and tested on enrollment images using cross-validation (down sampled).

Dataset	Baseline accuracy	Classification accuracy	Mean # of features	# of images with ties	# of images with 5-way tie
House 1	22.8%	98.4}	249	0	0
House 2	29.9%	77.4%	66	50	21
House 3	30.2%	96.9%	352	2	0
Workplace 1	24.4%	86.8%	31	133	52
Workplace 2	25.4%	93.5%	44	39	17
Average	26.5%	**90.6%**	—	—	

to 76.2% for House 2 (sparsely decorated with relatively few number of feature points), which outperform the baseline (majority class) by over 2.5 times.

Some images have few interest feature points because they are blurry photos or wall photos. Tables 5.3 shows the number of these images that have the same number of features across different rooms. In that case, the classifier uses random guessing between rooms to make the prediction.

τ is the main parameter which decides if the feature is discriminative or not. Larger values for τ increases the number of feature points during the matching process. Small values decrease the number of feature points, but keep more distinctive ones. We empiri-

cally found minimal sensitivity for τ between 0.3 and 0.6. For these experiments, we set τ to 0.45.

Global features. The main problem with the local classifier is that it fails on images with few distinctive points, because there are few feature matches and the classifier must resort to random guessing. Our global features are designed to address this problem by building models of general scene-level characteristics instead of local-level features. Table 5.5 compares classification performance of our six global features, using the same evaluation criteria as with the local features — 5-way classification using cross validation on the enrollment set. For the datasets with relatively few features, such as the sparsely-decorated House 2, the best global features outperform the local features (78.8% vs. 76.2% for House 2, and 93.9% vs. 84.0% for Workspace 1), but for the other sets the local features still dominate.

For final global features classifier we combine the two bags-of-SIFT and the bags-of-HOG features as our global features.

CNN features. Our model structure is very similar to BVLC Reference CaffeNet model [108]. We initialize our model parameters using a model pre-trained on the Places dataset [144], and then we used our enrollment datasets to fine-tune the model parameters. Table 5.6 shows the results of the global, local, CNN classifiers on single image classification task on our test datasets. CNN outperforms other features by almost 13%.

5.3.3.3 TEMPORAL STREAM CLASSIFICATION

We evaluate the proposed probabilistic joint image stream labeling technique. Here, we used all of the enrollment photos for training and used the photo streams for testing. We performed inference on the Hidden Markov Model (HMM) by using the Viterbi algorithm

Table 5.5: Global feature classifier trained and tested on enrollment images using cross-validation.

Dataset	Baseline accuracy	Bags of SIFT	Dense bags of SIFT	Bags of HOG	LBP	GIST	RGB histogram
House 1	22.8%	**89.1%**	81.4%	82.7%	41.6%	71.9%	57.4%
House 2	29.9%	49.7%	**78.8%**	78.7%	52.8%	64.8%	47.9%
House 3	32%	**89.4%**	68.9%	66.2%	51.9%	65.5%	57.4%
Workplace 1	24.4%	83.2%	**93.9%**	88.8%	76.2%	85.1%	79.8%
Workplace 2	25.4%	73.8%	83.1%	**83.2%**	67.5%	72.2%	55.0%
Average	26.5%	77.0%	**81.2%**	79.9%	58.0%	71.9%	59.5%

to find the most likely sequence of states given evidence from the entire image stream.

Table 5.6 shows the results of this step. When classifying single images, the global, local and classifiers perform roughly the same, except for the sparsely-decorated House 2 where global features outperform local features by almost eight percentage points. On average, the classifiers outperform a majority baseline classifier by almost 2.5 times.

The HMM provides a further and relatively dramatic accuracy improvement, improving average accuracy from 64.7% to 81.9% for local features, from 64.3% to 74.8% for global features, and from 77.67% to 86.05% for CNN as shown in Table 5.6 and Table 5.7. Combining the three types of features together with the HMM yields the best performance with an average accuracy of 93.48%, or over 3.1 times the baseline.

Figure 5.4 shows some sample images from the House 2 stream, including a random assortment of correctly and incorrectly classified images. We can speculate on the cause of some of the misclassifications. When images are collected looking through windows or doors such that little of an enrolled space is captured in the image, the classifier confidence is intuitively reduced. Similarly, high degrees of occlusion in images will frustrate classification attempts.

Human interaction. An advantage of our probabilistic approach is that it can naturally

Table 5.6: Classification of test streams by the single image classifiers.

Dataset	Baseline accuracy	Local features	Global features	CNN
House 1	29.8%	**52.79%**	48.3%	52.48%
House 2	31.0%	41.81%	49.12%	**68.68%**
House 3	20.9%	81.46%	79.95%	**91.16%**
Workplace 1	32.1%	75.92%	74.55%	**87.86%**
Workplace 2	28.9%	71.55%	69.36%	**88.18%**
Average	28.5%	64.71%	64.22%	**77.67%**

Table 5.7: Classification of test streams using variations of stream classifier.

Dataset	Baseline accuracy	Local features	Global features	CNN	Combined	Combined + Human interaction
House 1	29.8%	89.75%	60.55%	55.72%	82.92%	89.12%
House 2	31.0%	54.93%	56.68%	81.21%	85.91%	88.05%
House 3	20.9%	97.41%	89.87%	99.13%	99.56%	100.00%
Workplace 1	32.1%	75.53%	89.23%	95.30%	99.02%	99.80%
Workplace 2	28.9%	92.34%	81.40%	95.62%	100.00%	100%
Average	28.5%	81.99%	75.54%	85.30%	93.48%	95.39%

Figure 5.4: Some sample classification results from the House 2 stream, showing correctly classified (top) and incorrectly classified (bottom) images.

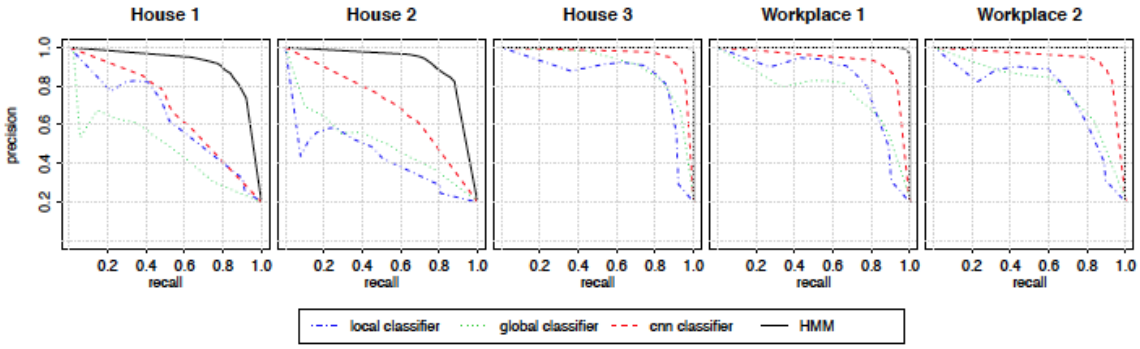


Figure 5.5: Precision-Recall curves for retrieving images from a given room, averaged over the five rooms, for each of our five datasets.

incorporate additional evidence. For example the user can interact with the PlaceAvoider system by labeling ambiguous images. We simulated this by having the HMM identify the least confident of its estimated labels and then we force these images to take the correct label with 1 probability, then we re-ran inference. We fix label for 10 images. Table 5.7 shows how human interaction improves the accuracy classification for all datasets.

Retrieving private images. The main objective of PlaceAvoider to filter out the images taken in potentially private rooms. We consider this as an image retrieval problem where the goal is to retrieve the private images from the stream. Although our classification algorithms achieve high performance, they are not perfect. Users can provide the system with

a confidence threshold to select between a highly conservative or a highly selective filter based on their preferences and the sensitivity of the rooms. Figure 5.5 shows precision-recall curves for retrieving private images from each of our five datasets. To generate these, we applied five retrieval tasks for each dataset, one for each room, and then averaged the resulting P-R curves together. For the local, global and CNN features we used the maximum value (across classes) of $P_L(l_i|I)$, $P_G(l_i|I)$, and $P_C(l_i|I_i)$ respectively as the free parameter (confidence), and for the HMM we used the maximum marginal (across classes) of $P(l_i|I_1, ..., I_m)$ computed by the Forward-Backward algorithm. We see that for House 3, Workplace 1, and Workplace 2, we can achieve 100% recall at almost 100.0% precision, meaning that all private images could be identified while removing only less than 0.05% of the harmless images. For House 1 and we can achieve about 90.0% precision and recall.

5.4 OBJECTAVOIDER

In the last section, we introduced PlaceAvoider to identify potentially sensitive images based on "where" the photos were taken, screening out images from locations such as bedrooms and bathrooms. While this approach works well in some situations, it does not examine "what" objects in the image may signal privacy violations. In this work we present ObjectAvoider, which uses computer vision algorithms to detect images with objects that are often sources of sensitive information, such as computer screens. Computer screens are more likely to have sensitive information for examples, emails, instant message, and personnel records. ObjectAvoider detects the presence of monitors and the running applications on the computer screens (e.g., emails).

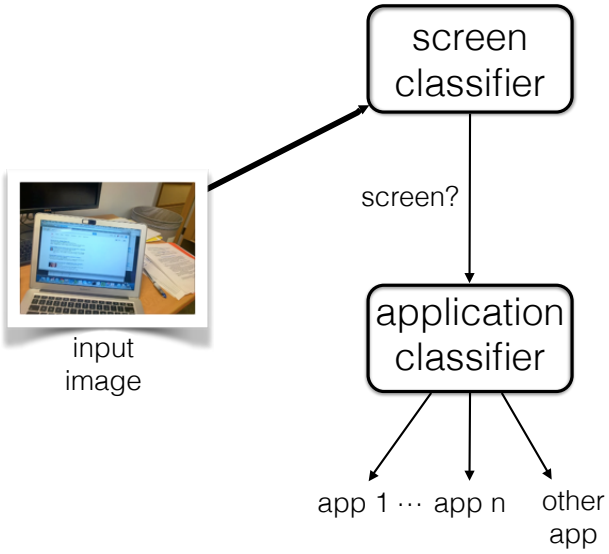


Figure 5.6: The ScreenAvoider hierarchical classifier. Native images are downsized for the Caffe CNN framework. While this depiction shows two classification levels, in Subsection 5.4.2.3 we also present a single classifier that includes applications and a class without screens.

5.4.1 SYSTEM ARCHITECTURE

Many current camera wearable devices (e.g Autographer, Narrative and Google Glass) provide cloud-based services for storing and managing the images. Our ObjectAvoider system in Figure 5.6 allows the organization of images based on their content. It is a hierarchical classifier, first classifying the image for presence of computer screens, and then if the image contains a monitor, it tests it further to check if it has any application of interest.

5.4.1.1 DETECTING COMPUTER SCREENS AND MONITORS IN IMAGES

To detect computer screens in images, we start by applying the traditional image classification approach to detect monitors in lifelogging images, using a set of traditional visual features. These include the features in Chapter 2, including simple image-level features like color histograms, more advanced scene layout features such as GIST [131] and Lo-

cal Binary Pattern histograms (which primarily capture global texture), and features that cue on local image regions including vector-quantized Histograms of Oriented Gradients (HOG) [37] and SIFT [97] features [128]. We then learned image classifiers with SVMs and thousands of annotated lifelogging images.

We also apply Convolutional Neural Networks to our problem of screen detection in lifelogging images. To our knowledge, no other work has studied CNNs with this type of data. As a reminder, one critical factor is that because the networks are so deep and thus have so many parameters, they need a very large training data to work correctly (and otherwise they "overfit" to a specific training set instead of learning general properties about it). We followed Oquab [108] et al. and started with a model pre-trained on the huge ImageNet dataset, even though that dataset has nothing to do with lifelogging or monitor detection. Using those network parameters as initialization, we then trained a network on monitor detection using our relatively small training dataset.

5.4.1.2 CLASSIFYING APPLICATIONS ON COMPUTER SCREENS

Detecting computers screens may be good enough for some applications. However, applying access control policies to restrict all images with screens may be too aggressive. Here, we are trying to discriminate between computer screen contents on the level of application. We consider three types of applications: (1) email applications, (2) social media websites, and (3) instant messenger services. The main objective for the system to identify the images with sensitive information. Due to the very good performance for CNNs, here we only consider the CNNs to discriminate between applications. Generally detecting the application within the screen is harder than detecting screens. In addition to deciding if there is screen in the image, the appearance of some websites is highly variable. Gmail offers

Table 5.8: A description of our datasets. The *irb study* dataset is an aggregation of images from 36 users. We collected *author* dataset using our lifelogging devices. The *flickr* images were manually scraped from Flickr and randomly sampled.

DataSet	Facebook	Gmail	Messenger	other	no monitor	total
irb study data	35	12	2	736	1957	2742
author	2750	2659	3046	3749	6594	18798
flickr	0	0	0	784	0	784
total	2785	2671	2799	5269	8551	22319

customized backgrounds which change its appearance, for example, while Facebook feeds look different for different users (e.g. ads). We test how well the classifier will generalize over these differences.

5.4.2 EVALUATION

We conducted a set of experiments to evaluate the performance of ObjectAvoider classifiers using different type of image data sets. We start by describing our evaluation datasets.

5.4.2.1 EVALUATION DATASETS

We could not find any public dataset that are suitable to evaluate ObjectAvoider. We used our lifelogging devices as the main source for collecting the data. We used a combination of Google Glass, Narrative Clip, Autographer, and lanyard worn smartphones with continuous photography applications. We manually labeled more than 18,000 images. Our IRB office was consulted and this effort was believed to not be human subjects research.

In addition to our dataset, we used a second dataset that was collected in situ by 36 participants in a human subject study [65]. We collected the last dataset from Flickr and manually labeled more than 784 images. These images are screenshots that contain monitor content. Table 5.8 describes our datasets in detail.

5.4.2.2 DETECTING COMPUTER SCREENS AND MONITORS

The main task for ObjectAvoider is to retrieve images with computer screens. To evaluate the performance on this task, we created three different experiments:

- ***Experiment Screen1*** - Train on 9,986 images from the *author* training partition. Test the model on 1,842 *author* images from the test partition that are randomly sampled such that there is an equal class distribution, so that a random classifier will achieve a baseline classification accuracy of 50%.
- ***Experiment Screen2*** - Train on 9,986 images from the *author* training partition. Test the model on all 2,742 *irb study data* images. 28.6% of these images have screens in them, which is the observed behavior from aggregating images from 36 users (so that a majority-class classifier will achieve a baseline accuracy of 71.4%).
- ***Experiment Screen3*** - Train on 9,986 images from the *author* training partition. Test the model on a mix of the 1,958 *irb* images without screens and 784 *flickr* images with screens. This experimental test set replaces the *irb* screen images with those scraped from Flickr (baseline remains 71.4%).

We trained a Convolutional Neural Network having 2.3M neurons with over 60M parameters. We used the BVLC Reference CaffeNet [69] model and modified the last layer. Table 5.9 shows the detailed configuration for our network. We initialize our models using the pre-trained model on the large ImageNet collection of Internet images.

Experiment Screen1 results - We consider this experiment as we provide upper-bound on the accuracy for retrieving images. In this experiments training and test images are sampled from the same photo streams which means there is a chance of similar images to appear in the two sets.

Table 5.9: BVLC Reference CaffeNet pre-trained model configuration with modification for ObjectAvoider. There are five sparsely connected convolutional layers and three fully connected layers that serve as a traditional neural network. Observe that only the last layer, fc3, changes with respect to the number of classes that are used. The parameter n is equal to the number of classes.

layer	# of filters	depth	width	height
data		3	227	227
conv1	96	3	11	11
conv2	256	48	5	5
conv3	384	256	3	3
conv4	384	192	3	3
conv5	256	192	3	3
fc1	1	4096	1	1
fc2	1	4096	1	1
fc3	1	n	1	1

Table 5.10: Experiment *Screen1* confusion matrix. Baseline is 50.0%. Accuracy is 99.8%.

		predicted	
		no screen	screen
actual	no screen	919	3
	screen	1	919

The CNN achieved 99.8% accuracy for this experiment. Table 5.10 contains the confusion matrix that shows only three false positives and one false negative. The incorrectly classified images are displayed in Figure 5.8. The false negative image is due to poor quality of the image, where information can be retrieved from the photographed screen. Figure 5.7 shows the retrieval performance for our classifier. Our classifier can recall 99% of screen images with 100% precision.

Experiment Screen2 results - In this experiment, the test and training datasets are completely independent. The training images are from the *author* dataset while the test from *irb* dataset. The test dataset is from the Hoyle et al. study [65], collected by 36 individuals

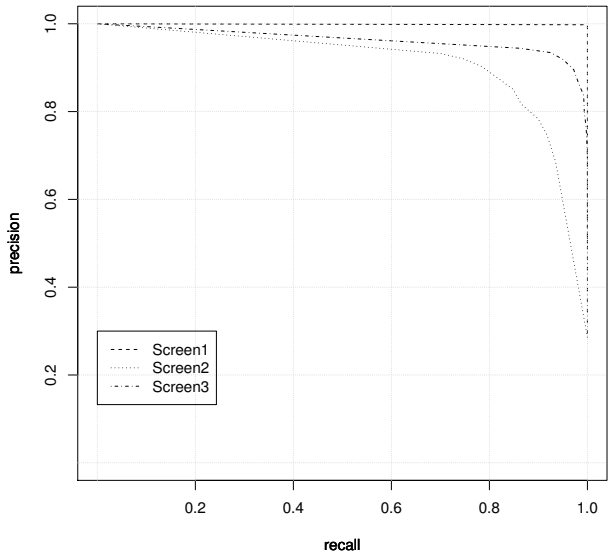


Figure 5.7: Precision and recall curves for retrieving images with computer screens.

in unconstrained settings. The class distribution in this case reflects the true distribution of monitors in the real-world study. The test images display much higher degrees of motion blur, noise, and poor exposure (highlights) due to the software and camera used in collecting the data [65].

Our CNN network demonstrated 91.5% accuracy for this experiment. Table 5.11 presents the confusion matrix. It shows an almost equal mix of false negative and false positive instances. We do not include samples from test images because of the IRB-controlled human subject study data rules. Instead, we reviewed all incorrectly classified images and report our observations from the manual reviewed. Table 5.12 shows an analysis of the 117 false negative images. 49.6% of the false negative images had computer screens present that were displaying video games in full screen mode while 12.8% of the images capture media in full screen mode (movies, sports, and television shows). It is worth mentioning that our training data has no examples for these types of images. Only 8 images (6.8%) from

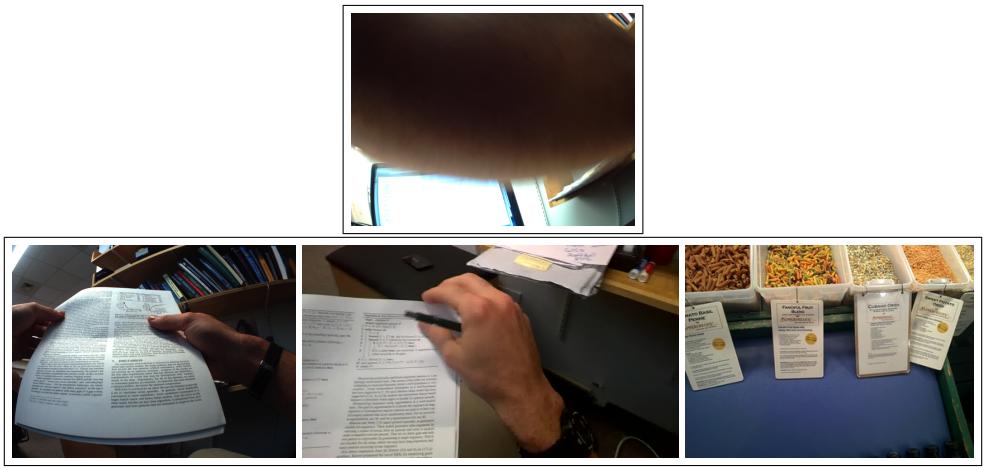


Figure 5.8: All four of the incorrectly classified Experiment *Screen1* photos (there were 1842 images in this test set). The top panel contains the only false negative case which is mostly occluded with the screen over-exposed. The bottom panel contains the three false positive cases.

false negative, 0.3% of the overall test images) contained sensitive content by a conservative definition (1 Skype screenshot, 2 Microsoft Word screenshots, 3 Facebook shots, and 2 Adobe Illustrator shots).

Similar to analysis of false negative images, we manually reviewed false positive images. Table 5.13 presents the results for false positive images. We found that the main reason came from images where windows or other framed objects were prominent. Also, 16.4% of the false positive images were screens of televisions, projectors, or smartphones instead of computers. Actually that shows the semantic power and the generalizability of deep learning techniques.

Figure 5.7 presents the precision recall curve for this expirement. Results are worst than the *screen1* experiment. Our classifier retrieve 88% of screen images with 80% precision.

Experiment Screen3 results - Our main objective in the last experiments is to check the ability of a classifier trained on one type of images to classify images of another type. This

Table 5.11: Experiment *Screen2* confusion matrix. Baseline is 71.4%. Accuracy is 91.5%

	predicted no screen	predicted screen
actual no screen	1842	117
actual screen	116	667

Table 5.12: Experiment *Screen2* false negative (FN) analysis. The FN images were manually reviewed and the following observations were made about the listed fraction of images. We speculate that these observed properties frustrated classification attempts. Note that these observation categories are not mutually exclusive.

	fraction of FN images
full screen video games	49.6%
less than 50% of screen visible	48.9%
significantly out of focus	35.0%
movie or TV show being played	12.8%
screen with sensitive information	0.3%

experiment is similar to *Screen2* in that they share the same negative class, but the positive class contains monitor images that are randomly collected from Flickr. Our classifier achieved 95.3% accuracy which is better than experiment *Screen2*. The confusion matrix are presented in Table 5.14. Figure 5.7 shows the precision recall curve for this experiment. Classifier can obtain recall 98% of screen images with 80% precision. This experiment further shows the ability to detect monitors in general.

5.4.2.3 CLASSIFYING APPLICATIONS

Here we test classifying images based on screen content. The main objective for this task is to allow users to share images containing screens that do not contain sensitive information. We conducted three experiments to evaluate the application classifier.

Table 5.13: Experiment *Screen2* false positive (FP) analysis. The FP images were manually reviewed and the following observations were made about the listed fraction of images. We speculate that these observed properties frustrated classification attempts. Note that these observation categories are not mutually exclusive.

	fraction of FP images
prominent window visible	33.6%
other *framed* element	32.8%
non-computer device with screen	16.4%

Table 5.14: Experiment *Screen3* confusion matrix. Baseline is 71.4%. Accuracy is 95.3%.

	predicted no screen	predicted screen
actual no screen	1842	117
actual screen	12	771

- ***Experiment App1*** - Binary classification between *sensitive* applications versus other applications. Train on 9,986 images from the *author* training partition. Test the model on 5,050 *author* images from the test partition that are randomly sampled such that there is an equal class distribution (baseline is 50%).

- ***Experiment App2*** - Four-way classification between Facebook, Gmail, Apple Messenger, and an "other" category. Train on 9,986 images from the *author* training partition. Test the model on 6,868 *author* test images sampled for an equal class distribution (baseline is 25%).

- ***Experiment App3*** - Five-way classification between no-screen, Facebook, Gmail, Apple Messenger, and an "other" application category. Train on 9,986 images from the *author* training partition. Test the model on all 2,742 *irb study data* images. 28.6% of these images have screens in them, which is the observed behavior from aggregating images from 36 users (baseline is 71.4%). The distribution of other applications is

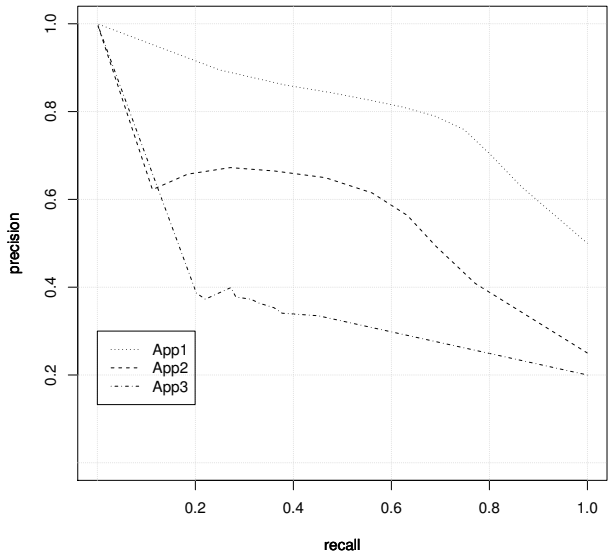


Figure 5.9: Precision and recall curves for the application classification experiments.

extremely unbalanced as shown in Table 5.16.

Here we trained different CNNs, one for each experiment. All the networks have the same configuration described in Table 5.9 except the last layer is modified based on the number of classes.

Experiment App1 results: In this experiment we try to classify application as a binary task, with one class for "sensitive application" which includes Facebook, Gmail and Apple Messenger. The other class include all screens showing any other application. Our classifier obtained 75.1% outperforms the random guessing classifier (the baseline 50%). Table 5.15 shows the confusion matrix and Figure 5.9 shows the precision-recall curve. The classifier can recall 80% of sensitive applications with 71% precision. Also, it is more biased toward false positives than false negative. That means the classifier is more likely to be restrictive by labeling "other applications" as sensitive than vice versa.

Table 5.15: Experiment *App1* confusion matrix. Baseline is 0.500. Accuracy is 0.751.

	predicted other app	predicted sensitive app
actual other app	1717	808
actual sensitive app	449	2076

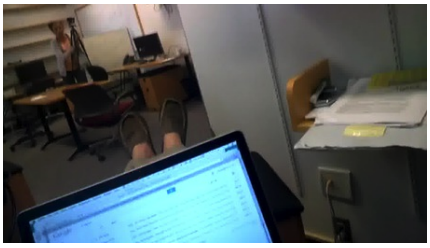
Experiment App2 results: We conducted this experiment to evaluate the performance of the classifier to discriminate amongst individual application. This fine-grain classification allow the users to share images from some application while preventing others. The random baseline in this case is 25%. Our classifier achieves 54.2%. Figure 5.10 contains an example image from each of the four categories that was classified correctly.

We carefully chose the representative applications in order to rigorously evaluate ObjectAvoider:

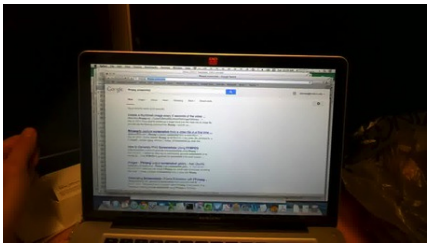
- Facebook displays a large degree of variation in visual content. Signature visual features (e.g., the blue banner) come and go depending on context. Much of the screen contains content personalized to the user.
- Gmail is an example of an email service that is browser-based and difficult to visually distinguish from other web content (especially other Google web services).
- Apple Messenger has a minimalist visual theme that was deliberately chosen as an example of a messenging application that is not easily recognizable.

It is clear that classifier's ability to discriminate amongst a given pair of applications is largely dependent on the choice of applications. Our evaluated applications and lifelogging datasets present challenging cases and we expect improved performance in the general case. Figure 5.9 shows the precision recall curve. The classifier can recall 80% of the desired images with a precision of less than 40%.

Experiment App3 results: The third experiment demonstrates more difficult conditions.

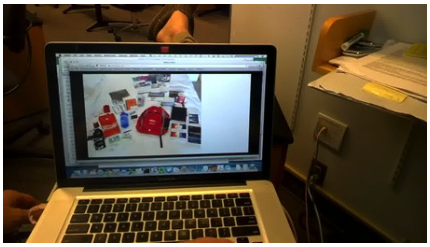


Gmail



other app (Google search)

Messenger



Facebook

Figure 5.10: Examples of images that were correctly classified in experiment *App2*. Note the ability of the classifier to discriminate amongst Google search and GMail which have similar visual features. The blue box is added for anonymity.

This is a five way classification problem where we have four application classes and a class represent no-screen images. While our *author* training data has reasonably balanced classes, the *irb study* test data for this experiment has a high degree of imbalance. The resulting accuracy for this experiment is 77.7% which is better than 71.4% baseline. Table 5.16 shows the confusion matrix. The classifier performs well at the coarse level of inferring whether or not a screen is present, but classification amongst sensitive applications is very poor. Figure 5.9 show the precision recall curve for this classifier. This classifier can retrieve 80% of desired images with a precision of about 25%.

Other application classification approaches - We also tried other experiments outside of the three that we detail above. For example we considered using CNN-generated features with a different choice of classifier (e.g. SVMs and Decision Trees). We applied two models to extract the features: the standard BVLC Reference CaffeNet pre-trained model and the

Table 5.16: Experiment *App3* confusion matrix. Baseline is 71.4%. Accuracy is 77.7%.

	predicted no screen	predicted other app	predicted messenger	predicted facebook	predicted gmail
actual no screen	1882	59	6	0	12
actual other app	157	243	143	35	158
actual messenger	0	2	0	0	0
actual facebook	4	12	11	5	3
actual gmail	0	7	3	0	2

fined-tuned model based on our data set. However, CNN classifier outperformed all these approaches.

5.5 SUMMARY

In this chapter we presented two systems based on image classification and machine learning for screening images to help first-person camera users to maintain their privacy:

- PlaceAvoider [128] analyzes images to determine where they were taken, and to filter out images from places like bedrooms and bathrooms, and
- ObjectAvoider [80] filters images based on their content, looking for objects that may signal privacy violations (e.g., computer monitors).

Both systems use a combination of traditional visual features and deep learning techniques to classify images. PlaceAvoider detects a potentially sensitive images taken from first-person cameras by recognizing physical areas where sensitive images are likely to be captured. ObjectAvoider tries to detect images with monitors and further tries to discriminate between the content of the monitors based on the running applications.

We presented different sets of evaluation experiments for both systems. Our results show the effectiveness of the proposed systems, and that deep learning techniques outperform traditional visual features by more than 10% in both systems. These systems can

be seen as first steps toward the larger goal of detecting sensitive images in first-person cameras.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In this thesis we have studied image classification methods for real world applications. We presented two lines of work: one for mining photo-sharing social media and the second for maintaining the privacy of first-person cameras.

The main contributions of this thesis are:

- We present image-classification systems utilizing image classification and deep learning for unconstrained, realistic, and automatically collected data sets, that include methods for modeling and removing noisy data in social media and first person images.
- We present novel applications of image classification for ecology and privacy.
- We build realistic data sets in these two domains.

In Chapter 2, we introduced image classification and described image classification components and deep learning technology, and then Chapter 3 showed a straightforward image classification example that can solve avatar image Captchas. Our results showed that avatar captchas are not secure, because modern image classification techniques can solve them very accurately.

In Chapter 4, we presented an image classification system to mine the massive collections of user generated photos uploaded to social media websites as a source of observational evidence about the natural world, and in particular as a way of estimating the presence of ecological phenomena. Our work is an initial step towards a long-term goal of monitoring important ecological events and trends through online social media. Our system has two major components: image classification and a probabilistic model. In the image classification component, we applied visual features and deep learning to classify the images. Then, we presented a probabilistic model (Bayesian likelihood ratio) to deal with noisy and biased data in social media. Our proposed model combines the outputs from image classification in a principled probabilistic way to make final predictions about the phenomenon. We evaluated our methods on large scale unconstrained data set consist of hundreds of millions of images, and in particular, we used a collection of more than 200 million geo-tagged, timestamped photos from Flickr to estimate snow cover and greenery, and compared these estimates to fine-grained ground truth collected by earth-observing satellites and ground stations. Our results showed that while the recall is relatively low due to the sparsity of photos on any given day, the precision can be quite high, suggesting that mining from photo sharing websites could be a reliable source of observational data for ecological and other scientific research.

In Chapter 5, we presented two systems to help maintaining the privacy of first person camera. Our systems help users to manage images collected from wearable devices. The first system is PlaceAvoider which tries to detect sensitive images based on where they were taken. PlaceAvoider also has two main parts: image classification and a probabilistic model. We use three different classifiers to classify the images into indoor scene categories. We use local, global and deep learning classifiers to predict the location of the image. Then

we proposed a Hidden Markov Model as stream classifier to benefit from the temporal information encoded in the life-logging images. We collected different data sets to evaluate our proposed system, and our results show the high performance for the proposed system. The second proposed system is ObjectAvoider which tries to detect and protect computer screens of lifeloggers. Our proposed system also tries to detect screen and sensitive applications running on the screen. We used a data set collected from 36 lifelogging users to evaluate our system. Our results showed that policies based on the detection of computer screens in first person images could be applied at a coarse level very accurately. However, fine-grained policies that based on detecting types of computer screen content are more challenging to enforce. Our results in that direction are optimistic, especially when discriminating sensitive vs. non-sensitive applications.

6.2 FUTURE WORK

In applying image classifications for ecology, we plan to study a variety of other ecological phenomena, including those for which high quality ground truth is not available, such as migration patterns of wildlife and the distributions of blooming flowers. Other possibilities for future work are to develop more sophisticated techniques for dealing with noisy and biased image data in social media. We plan to collect a large data set of different natural scenes, then build and train a Convolutional Neural Network designed for this specific purpose. Generally, we hope the idea of observing nature through photo-sharing websites will help spark renewed interest in recognizing natural and ecological phenomenon in consumer images.

In applying image classification for privacy, there is a potential challenge to investigate more techniques that estimate meaning in images to better identify potentially sensitive

photo content and situations. We hope this work starts to bring attention to this area, eventually leading to automated systems that decrease the labor managing wearable camera imagery and at the same time maintain the privacy of first person cameras.

Besides privacy and ecology, there are many interesting and challenging applications for image classification that need to be explored at a large scale. For instance, visual sentiment analysis tries to analyze emotion, affect and sentiment from visual content. Visual sentiment analysis is more challenging than other image classification tasks including object recognition and scene categorization as the latter problems are well-defined while sentiment analysis is more subjective and requires a higher level of abstraction. One interesting direction is to combine visual and textual evidence to build powerful large scale sentiment systems for social media using deep learning.

Deep learning has advanced the state-of-the-art in different fields including vision. Deep learning is not a new technology, as basic building blocks such as CNNs have been around for decades [17, 85, 86, 88, 89, 102], but the increase in amount of the data and computational power brought them back to the top of the state-of-the art. One interesting direction to improve deep learning is to build methods to learn the structure and hyper parameters of the networks. This is still not an explored direction due to the complex structure of the networks and the complicated process of training these networks. Deep learning is inspired by biological models which open the door for many other methods that are inspired from biology but were neglected in the computer vision community. For example, genetic algorithms [51] and artificial immune systems [38] are powerful optimization techniques that could be used in the vision community due to the current advances of computational power (e.g. GPU) and the enormous amount of available data. Genetic algorithms are inspired by the process of natural selection while artificial immune

systems (AIS) are inspired by the principles and processes of biological immune systems. Both of these systems worth exploring in the context of image classification systems at a large scale.

In this thesis, we presented image classification systems which operate in the unconstrained context of real world applications for which the metric of success is their usefulness. While there is always room to improve feature representations and classifier performance (especially in egocentric images), we believe that the current state of the art is sufficient to enable interesting and essential capabilities for the scientific community to harness large-scale image sources.

BIBLIOGRAPHY

[1] http://blog.flickr.net/en/2014/11/24/science-and-flickr-powering-product-yahoo-weather/.

[2] http://www.greatsunflower.org.

[3] http://www.ncdc.noaa.gov/oa/climate/ghcn-daily/.

[4] Hand written recognition. http://yann.lecun.com/exdb/lenet/, (accessed March. 30, 2015).

[5] Medical image. http://www.clarontech.com/clinical-appdev.php, (accessed March. 30, 2015).

[6] Retail application. http://www.datalogic.com/eng/products/, (accessed March. 30, 2015).

[7] Surveillance and security applications. http://visionwang.com/2008/12/07/mobile-vision-iphone-apps-employ-computer-vision-and-image-processing-techs/, (accessed March. 30, 2015).

[8] C. Aggarwal and T. Abdelzaher. Social Sensing. In *Managing and Mining Sensor Data*. Springer, 2013.

[9] L. Ahn, M. Blum, N. Hopper, and J. Langford. CAPTCHA: Using hard AI problems for security. In *Proceedings of the International Conference on Theory and Applications of Cryptographic Techniques*, pages 294–311, 2003.

[10] A. Allen. Dredging up the past: Lifelogging, memory, and surveillance. *The University of Chicago Law Review*, pages 47–74, 2008.

[11] A. Almazyad, Y. Ahmad, and S. Kouchay. Multi-modal captcha: A user verification scheme. In *Information Science and Applications (ICISA), 2011 International Conference on*, pages 1–7. IEEE, 2011.

[12] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 7–15. ACM, 2008.

[13] *Autographer Wearable Camera*, (accessed Jul. 5, 2013). `http://autographer.com`.

[14] P. Balamurugan and R. Rajesh. Greenery image and non-greenery image classification using adaptive neuro-fuzzy inference system. In *International Conference on Computational Intelligence and Multimedia Applications.*, volume 3, pages 431–435. IEEE, 2007.

[15] P. C. Barnum, S. Narasimhan, and T. Kanade. Analysis of rain and snow in frequency space. *International Journal of Computer Vision*, 86(2-3):256–274, 2010.

[16] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *European Conference on Computer Vision*, pages 404–417, 2006.

[17] Y. Bengio, Y. LeCun, and D. Henderson. Globally trained handwritten word recognizer using spatial representation, convolutional neural networks, and hidden

markov models. *Advances in Neural Information Processing Systems*, pages 937–937, 1994.

[18] J. Bollen, H. Mao, and X.-J. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[19] F. Bonin-Font, A. Ortiz, and G. Oliver. Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3):263–296, 2008.

[20] M. Boutell. *Exploiting context for semantic scene classification*. PhD thesis, University of Rochester Department of Computer Science, 2005.

[21] M. Boutell, A. Choudhury, J. Luo, and C. M. Brown. Using semantic features for scene classification: How good do they need to be? In *IEEE International Conference on Multimedia and Expo.*, pages 785–788. IEEE, 2006.

[22] D. Brockmann, L. Hufnagel, and T. Geisel. The scaling laws of human travel. *Nature*, 439:462–465, 2006.

[23] K. Caine. Supporting privacy by preventing misclosure. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 3145–3148. ACM, 2009.

[24] A. Chandavale, A. Sapkal, and R. Jalnekar. A framework to analyze the security of text-based CAPTCHA. *International Journal of Computer Applications*, 1(27):127–132, 2010.

[25] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. on Intelligent Systems and Technology*, 2:1–27, 2011.

[26] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *Proceedings of the British Machine Vision Conference*, 2011.

[27] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[28] J. Chaudhari, S. Cheung, and M. Venkatesh. Privacy protection for life-log video. In *IEEE Workshop on Signal Processing Applications for Public Security and Forensics*, pages 1–5, 2007.

[29] Y. Chen and G. J. F. Jones. Augmenting human memory using personal lifelogs. In *Proceedings of the 1st Augmented Human International Conference*, AH '10, pages 24:1–24:9, New York, NY, USA, 2010. ACM.

[30] W. Cheng, L. Golubchik, and D. Kay. Total recall: are privacy changes inevitable? In *ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*, pages 86–92, 2004.

[31] Y. Chon, N. Lane, F. Li, H. Cha, and F. Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. In *ACM Conference on Ubiquitous Computing*, pages 481–490, 2012.

[32] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 160–168. ACM, 2008.

[33] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3001–3008. IEEE, 2011.

[34] D. Crandall and N. Snavely. Networks of landmarks, photos, and people. *Leonardo*, 44(3):240–243, 2011.

[35] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web*, pages 761–770. ACM, 2009.

[36] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision at the European Conference on Computer Vision (ECCV)*, 2004.

[37] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[38] D. DasGupta. *Artificial immune systems and their applications*. Springer Publishing Company, Incorporated, 2014.

[39] B. De Longueville, R. S. Smith, and G. Luraschi. "OMG, from here, i can see the flames!". In *Proc. International Workshop on Location Based Social Networks*, pages 73–80, 2009.

[40] T. Denning, Z. Dehlawi, and T. Kohno. In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies. In *International Conference on Human Factors in Computing Systems*, 2014.

[41] M. Douze, H. Jegou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of GIST descriptors for web-scale image search. In *ICIVR*, 2009.

[42] D. D'Souza, P. Polina, and R. Yampolskiy. Avatar captcha: Telling computers and humans apart via face classification. In *IEEE International Conference on Electro/Information Technology (EIT)*, pages 1–6. IEEE, 2012.

[43] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[44] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[45] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[46] D. Fink, T. Damoulas, and J. Dave. Adaptive spatio-temporal exploratory models: Hemisphere-wide species distributions from massively crowdsourced eBird data. In *AAAI*, 2013.

[47] G. D. Forney Jr. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[48] J.-M. Frahm, P. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, and S. Lazebnik. Building Rome on a Cloudless Day. In *European Conference on Computer Vision*, 2010.

[49] H. Gao, D. Yao, H. Liu, X. Liu, and L. Wang. A novel image based CAPTCHA using jigsaw puzzle. In *IEEE International Conference on Computational Science and Engineering (CSE)*, pages 351–356. IEEE, 2010.

[50] J. Ginsberg, M. Mohebbi, R. Patel, L. Brammer, M. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457:1012–1014, 2009.

[51] D. E. Goldberg. *Genetic algorithms*. Pearson Education India, 2006.

[52] S. A. Golder and M. Macy. Diurnal and Seasonal Mood Vary with Work, Sleep, and Daylength Across Diverse Cultures. *Science*, 333(6051):1878–1881, Sept. 2011.

[53] D. Grangier, L. Bottou, and R. Collobert. Deep convolutional networks for scene parsing. In *ICML 2009 Deep Learning Workshop*, volume 3, 2009.

[54] W. E. L. Grimson and J. L. Mundy. Computer vision applications. *Communications of the ACM*, 37(3):44–51, 1994.

[55] D. Hall, G. Riggs, and V. Salomonson. MODIS/Terra Snow Cover Daily L3 Global 0.05Deg CMG V004. National Snow and Ice Data Center, updated daily.

[56] M. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[57] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[58] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

[59] C. Hauff and G.-J. Houben. Geo-location estimation of Flickr images: social web based enrichment. In *Advances in Information Retrieval*, pages 85–96. Springer, 2012.

[60] J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008.*, pages 1–8. IEEE, 2008.

[61] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, Aug. 2001.

[62] T. K. Ho. Nearest neighbors in random subspaces. In *Advances in Pattern Recognition*, pages 640–648. Springer, 1998.

[63] S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. Wood. Sensecam: a retrospective memory aid. In *ACM Conference on Ubiquiotous Computing*, 2006.

[64] R. Hoyle, R. Templeman, D. Anthony, D. Crandall, and A. Kapadia. Sensitive lifelogs: A privacy analysis of photos from wearable cameras. In *The ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '15)*, 2015.

[65] R. Hoyle, R. Templeman, S. Armes, D. Anthony, D. Crandall, and A. Kapadia. Privacy behaviors of lifeloggers using wearable cameras. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 571–582. ACM, 2014.

[66] O. Hyvarinen and E. Saltikoff. Social media as a source of meteorological observations. *Monthly Weather Review*, 138(8):3175–3184, 2010.

[67] *iRON Wearable Camera*, (accessed March. 30, 2015). http://usa.ioncamera.com/snapcam/.

[68] S. Jana, A. Narayanan, and V. Shmatikov. A Scanner Darkly: Protecting user privacy from perceptual applications. In *34th IEEE Symposium on Security and Privacy*, 2013.

[69] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013.

[70] X. Jin, A. Gallagher, L. Cao, J. Luo, and J. Han. The wisdom of social multimedia: Using Flickr for prediction and forecast. In *ACM Multimedia*, 2010.

[71] G. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 338–345, 1995.

[72] C. Kanan and G. Cottrell. Robust classification of objects, faces, and flowers using natural image statistics. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[73] K. Kang, Y. Kwon, Y. Kim, J. Lee, and C. Bae. Lifelog collaboration framework for healthcare service on android platform. In *ICT for Smart Society (ICISS), 2013 International Conference on*, pages 1–4. IEEE, 2013.

[74] S. Kantabutra. Vision effects in thai retail stores: practical implications. *International Journal of Retail & Distribution Management*, 36(4):323–342, 2008.

[75] T. Karkkainen, T. Vaittinen, and K. Vaananen-Vainio-Mattila. I don't mind being logged, but want to remain in control: a field study of mobile activity and context logging. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 163–172, 2010.

[76] J. King, A. Cabrera, and R. Kelly. The Snowtweets Project: Communicating snow depth measurements from specialists and non-specialists via mobile communication technologies and social networks. In *AGU Fall Meeting*, 2009.

[77] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[78] M. Korayem, A. Mohamed, D. Crandall, and R. Yampolskiy. Learning visual features for the avatar captcha recognition challenge. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 2, pages 584–587. IEEE, 2012.

[79] M. Korayem, A. A. Mohamed, D. Crandall, and R. V. Yampolskiy. Solving avatar captchas automatically. In *Advanced Machine Learning Technologies and Applications*, pages 102–110. Springer, 2012.

[80] M. Korayem, R. Templeman, D. Chen, D. Crandall, and A. Kapadia. Screenavoider: Protecting computer screens from ubiquitous cameras. *arXiv preprint arXiv:1412.0008*, 2014.

[81] K. Kremerskothen. http://blog.flickr.net/en/2011/08/04/6000000000/.

[82] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[83] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *European Conference on Computer Vision*, pages 502–516. Springer, 2012.

[84] Land Processes Distributed Active Archive Center. MODIS/Terra Vegetation Indices 16-Day L3 Global 0.05Deg CMG V005. Sioux Falls, SD: U.S. Geological Survey, 2011.

[85] S. Lawrence, C. L. Giles, and A. C. Tsoi. Convolutional neural networks for face recognition. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 217–222. IEEE, 1996.

[86] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural-network approach. *Neural Networks, IEEE Transactions on*, 8(1):98–113, 1997.

[87] D. Lazer et al. Life in the network: the coming age of computational social science. *Science*, 323(5915):721–723, 2009.

[88] Y. LeCun and Y. Bengio. Word-level training of a handwritten word recognizer based on convolutional neural networks. In *International Conference on Pattern Recognition*, pages 88–88, 1994.

[89] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361:310, 1995.

[90] D. Leung and S. Newsam. Proximate Sensing: Inferring What-Is-Where From Geo-referenced Photo Collections. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[91] L.-J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. In *IEEE 11th International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.

[92] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2036–2043. IEEE, 2009.

[93] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs. In *European Conference on Computer Vision*, pages 427–440, 2008.

[94] Y. Li, D. Crandall, and D. P. Huttenlocher. Landmark Classification in Large-scale Image Collections. In *IEEE International Conference on Computer Vision*, 2009.

[95] D. Liben-Nowell and J. Kleinberg. Tracing information flow on a global scale using internet chain-letter data. *Proceedings of the National Academy of Sciences*, 105(12):4633–4638, 2008.

[96] D. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, pages 1150–1157, 1999.

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

[98] J. Luo, A. Singhal, and W. Zhu. Natural object detection in outdoor scenes based on probabilistic spatial context models. In *International Conference on Multimedia and Expo. ICME'03.*, 2003.

[99] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical image analysis*, 1(2):91–108, 1996.

[100] *Memoto Lifelogging Camera*, (accessed Jul. 5, 2013). `http://memoto.com`.

[101] C. Murdock, N. Jacobs, and R. Pless. Webcam2Satellite: Estimating cloud maps from webcam imagery. In *Applications of Computer Vision (WACV)*, pages 214–221. IEEE, 2013.

[102] C. Nebauer. Evaluation of convolutional neural networks for visual recognition. *Neural Networks, IEEE Transactions on*, 9(4):685–696, 1998.

[103] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[104] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 722–729. IEEE, 2008.

[105] D. Noyes. Facebook statistics. https://zephoria.com/social-media/top-15-valuable-facebook-statistics/, (accessed March. 30, 2015).

[106] B. O'Connor, R. Balasubramanyan, B. Routedge, and N. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *The International AAAI Conference on Web and Social Media (ICWSM)*, 2010.

[107] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[108] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1717–1724. IEEE, 2014.

[109] M. L. Parry, O. F. Canziani, J. P. Palutikof, P. J. van der Linden, and C. E. Hanson. *IPCC, 2007: Climate Change 2007: Impacts, Adaptation, and Vulnerability*. Cambridge University Press, 2007.

[110] S. Prabhakar, S. Pankanti, and A. K. Jain. Biometric recognition: Security and privacy concerns. *IEEE Security & Privacy*, 1(2):33–42, 2003.

[111] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[112] 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. arXiv:1409.0575, 2014.

[113] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

[114] A. Sadilek, H. Kautz, and V. Silenzio. Predicting Disease Transmission from Geo-Tagged Micro-Blog Data. In *Twenty-Sixth Conference on Artificial Intelligence (AAAI-12)*, 2012.

[115] K. Sage and S. Young. Security applications of computer vision. *Aerospace and Electronic Systems Magazine, IEEE*, 14(4):19–29, 1999.

[116] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th International Conference on the World Wide Web*, pages 851–860. ACM, 2010.

[117] H. K. Sarohi and F. U. Khan. Image retrieval using classification based on color. *International Journal of Computer Applications*, 64(11), 2013.

[118] A. Savvides, C. Han, and M. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *International Conference on Mobile Computing and Networking*, pages 166–179, 2001.

[119] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758, 2002.

[120] S. Se, D. Lowe, and J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.

[121] C. Siagian and L. Itti. Comparison of GIST models in rapid scene categorization tasks. *Journal of Vision*, 8(6):734–734, 2008.

[122] V. Singh, S. Venkatesha, and A. K. Singh. Geo-clustering of images with missing geotags. In *Granular Computing (GrC), 2010 IEEE International Conference on*, pages 420–425. IEEE, 2010.

[123] V. K. Singh, M. Gao, and R. Jain. Social pixels: genesis and evaluation. In *ACM Multimedia*, 2010.

[124] A. Singhal, J. Luo, and W. Zhu. Probabilistic spatial context models for scene content understanding. In *Computer Vision and Pattern Recognition (CVPR)*, 2003.

[125] N. Snavely, S. Seitz, and R. Szeliski. Modeling the World from Internet Photo Collections. *International Journal of Computer Vision*, 80:189–210, 2008.

[126] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[127] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[128] R. Templeman, M. Korayem, D. Crandall, and A. Kapadia. Placeavoider: Steering first-person cameras away from sensitive spaces. In *Network and Distributed System Security Symposium (NDSS)*, 2014.

[129] R. Templeman, Z. Rahman, D. Crandall, and A. Kapadia. PlaceRaider: Virtual theft in physical spaces with smartphones. In *Network and Distributed System Security Symposium*, 2013.

[130] B. Thomee, J. G. Moreno, and D. A. Shamma. Who's time is it anyway?: Investigating the accuracy of camera timestamps. In *Proceedings of the ACM International Conference on Multimedia*, pages 909–912. ACM, 2014.

[131] A. Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2):169–191, 2003.

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

[133] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[134] K. Truong, S. Patel, J. Summet, and G. Abowd. Preventing camera recording by designing a capture-resistant environment. In *International Conference on Ubiquitous Computing*, pages 73–86, 2005.

[135] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *IEEE International Conference on Robotics and Automation*, pages 1023–1029, 2000.

[136] A. Vailaya, M. A. Figueiredo, A. K. Jain, and H.-J. Zhang. Image classification for content-based indexing. *Image Processing, IEEE Transactions on*, 10(1):117–130, 2001.

[137] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *Proceedings of the 7th European Conference on Computer Vision*, pages 255–271. Springer, 2002.

[138] I. Vergara, T. Norambuena, E. Ferrada, A. Slater, and F. Melo. StAR: a simple tool for the statistical comparison of ROC curves. *BMC Bioinformatics*, 9(1):265, 2008.

[139] J. Wang, M. Korayem, and D. J. Crandall. Observing the natural world with Flickr. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 452–459. IEEE, 2013.

[140] L. Wang, X. Chang, Z. Ren, H. Gao, X. Liu, and U. Aickelin. Against spyware using captcha in graphical password scheme. In *IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pages 760–767. IEEE, 2010.

[141] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Real-Life Images workshop at the European Conference on Computer Vision (ECCV)*, October 2008.

[142] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3492, 2010.

[143] H. Zhang, M. Korayem, D. J. Crandall, and G. LeBuhn. Mining photo-sharing websites to study ecological phenomena. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 749–758, New York, NY, USA, 2012. ACM.

[144] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495, 2014.

Curriculum Vitae

Mohammed Korayem

EDUCATION

Ph.D. Computer Science, Indiana University, 2015. (Advisor: Prof. David J. Crandall)

M.S. Computer Science, Indiana University, 2013.

M.S. Computer Science, Cairo University, Egypt, 2009.

B.S. Computer Science, Cairo University, Egypt, 2002.

RESEARCH INTERESTS

Computer Vision and Data mining. I am particularly interested in visual and textual mining of large scale data.

EMPLOYMENTS

2010-2015 Research Assistant, Computer Science, School of Informatics and Computing, Indiana University, IN USA.

May-Aug 2014 Data Scientist Intern, CareerBuilder.

2012-2013 Teaching Assistant, School of Library and Information Science, Indiana University, IN USA.

2009-2010 Senior Software Engineer, Engineering for Integrated Systems (EIS), Egypt.

2005-2010 Teaching Assistant, Computer Science Dept., Fayoum University, Egypt.

2008-2009 Instructor(part-time), Information Technology Institute (ITI, Ministry of Communications and Information Technology, Egypt.

2003-2005 Teaching Assistant, Mathematics and Computer Science Dept., Cairo University, Egypt.

SELECTED PUBLICATIONS

Robert Templeman, Mohammed Korayem, David Crandall, and Apu Kapadia, **PlaceAvoider: Steering first-person cameras away from sensitive spaces**, in Proceedings of the 21st Annual Network and Distributed System Security symposium, 2014

Khalifeh Aljadda, Mohammed Korayem, Camilo Ortiz, Trey Grainger, John Miller, and William York, **PGMHD: A Scalable Probabilistic Graphical Model for Massive Hierarchical Data Problems**, in Proceedings of IEEE BigData, 2014.

Khalifeh Aljadda, Mohammed Korayem, Trey Grainger, and Chris Russell, **Crowdsourced Query Augmentation through Semantic Discovery of Domain-specific Jargon**, in Proceedings of IEEE BigData, 2014.

Mohammed Korayem and David J Crandall , **De-anonymizing users across heterogeneous social computing platforms**, in Proceedings of the International AAAI Conference on Weblogs and Social Media, 2013.

Jingya Wang, Mohammed Korayem, and David J Crandall, **Observing the natural world through Flickr**, in Proceedings of the The First IEEE International Workshop on Computer Vision for Converging Perspectives (in conjunction with ICCV 2013), **(Best Paper award)**

Haipeng Zhang, Mohammed Korayem, Erkang You, and David J. Crandall, **Beyond Co-occurrence: Discovering and Visualizing Tag Relationships from Geo-spatial and Temporal Similarities**, in Proceedings of the fifth ACM International Conference on Web Search and Data Mining, 2012.

Haipeng Zhang, Mohammed Korayem, David J. Crandall, and Gretchen Lebuhn, **Mining Photo-sharing Websites to Study Ecological Phenomena**, in Proceedings of the 21st International Conference on World Wide Web, 2012.

Mohammed Korayem, Abdallah A Mohamed, David Crandall, and Roman V Yampolskiy. **Learning visual features for the Avatar Captcha recognition challenge**, in Proceedings of the 11th International Conference on Machine Learning and Applications, 2012.

Mohammed Korayem, David Crandall, and Muhammad Abdul-Mageed, **Subjectivity and sentiment analysis of Arabic: A survey**, in Proceedings of the Advanced Machine Learning Technologies and Applications, Springer Berlin Heidelberg, 2012.

Muhammad Abdul-Mageed, Mona T Diab, and Mohammed Korayem, **Subjectivity and sentiment analysis of modern standard arabic**, in The Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL), short papers-Volume, 2011.

Muhammad Abdul-Mageed, Mohammed Korayem, and Ahmed YoussefAgha, **"Yes we can?": Subjectivity annotation and tagging for the health domain**, in The Proceedings of Recent Advances in Natural Language Processing (RANLP), 2011.

Mohamed Korayem, Amr Badr, and Ibrahim Farag, **Optimizing hidden markov models using genetic algorithms and artificial immune systems**, Computing and Information Systems, 11(2):44, 2007.

PATENTS

Robert Templeman, David Crandall, Apu Kapadia, and Mohammed Korayem, **A method and system of enforcing privacy policies for mobile sensory devices**, US patent pending.