Doctoral Thesis

A Study on Public Safety Prediction Using Satellite Imagery and Open Data

NAJJAR Al-Ameen

Laboratory of Information Communication Networks,
Graduate School of Information Science and Technology,
Hokkaido University

February 15, 2017

Contents

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

NAJJAR Al-Ameen

February 2017

Acknowledgments

I would like to sincerely thank my supervisor, Prof. Yoshikazu Miyanaga, from the Graduate School of Information Science and Technology, Hokkaido University, for the invaluable guidance in writing this thesis.

I would also like to sincerely thank Prof. Shun'ichi Kaneko, from the Graduate School of Information Science and Technology, Hokkaido University, for the countless hours of assistance and fruitful discussion over the course of performing the work described in this thesis.

Furthermore, I would like to sincerely thank everyone at the Laboratory of Information Communication Networks and the Laboratory of Human Centric Engineering, Graduate School of Information Science and Technology, Hokkaido University for their invaluable support and assistance.

Finally, I would like to thank the Ministry of Education, Culture, Sports Science and Technology, Japan, for the opportunity to study in Japan on a government scholarship.

Abstract

Data-driven public safety mapping is critical for the sustainable development of cities. Maps visualize patterns and trends about cities that are difficult to spot in data otherwise. For example, a road-safety map made from years' worth of traffic-accident reports pinpoints roads and highways vulnerable to accidents. Similarly, a crime map highlights where within the city criminal activities abound. Such insights are invaluable to inform sustainable city-planning decision-making and policy. Therefore, public-safety mapping is crucial for urban planning and development worldwide.

However, accurate mapping requires longitudinal data collection, which is both highly expensive and labor intensive. Data collection is manual and requires skilled enumerators to conduct. While rich countries are flooded with data, most of poor countries suffer from data poverty. Therefore, city-scale public safety mapping is beyond affordable to low- and middle-income countries. Thus, taking manual data collection out of the equation will quicken the mapping process in general, and make it possible where it is not.

Recent advances in imaging and space technology have made high-resolution satellite imagery increasingly abundant, affordable and more accessible. Satellite imagery has a bird's-eye/aerial viewpoint which makes it a rich medium of visual cues relevant to environmental, social, and economic aspects of urban development. Given the recent breakthroughs made in the field of computer vision and pattern recognition, it is straightforward to attempt predicting public safety directly from satellite imagery. In other words, investigating the use of visual information contained in satellite imagery as a proxy indicator of public safety.

In this study, we discuss our approach to public safety prediction directly from raw satellite imagery using tools from modern machine learning and computer vision. Our approach is applied at a city scale thus allowing for the automatic generation of city-scale public safety maps. In this work

we focus our attention on two types of public safety maps, namely road safety maps and crime maps. We formalize the problem of public safety mapping as a supervised image classification problem, in which a city-scale satellite map is treated as a set of satellite images each of each is assigned a safety label predicted using a model learned from training samples. To obtain this training data we leverage official police reports collected by police departments and released as open data. The idea is to mine large-scale datasets of official police reports for high-resolution satellite images labeled with safety scores calculated based on number and severity/category of incidents. We validate and test the robustness of the learned models for both road safety and crime rate prediction tasks over four different US cities, namely New York, Chicago, San Francisco, and Denver. We also attempt to investigate the reusability of the learned computational models across different cities.

This thesis consists of 5 chapters. Chapter 1 discusses both motivation and background of the study. It also describes how this thesis is organized. Chapter 2 overviews the contributions made in this study which can be summarized as follows: (1) proposing a framework for automatic city-scale public safety prediction from satellite imagery, (2) proposing an automatic approach for obtaining labeled satellite imagery via mining large-scale collections of official police reports released as open data, and (3) introducing five labeled satellite imagery datasets representing four different US cities, and mined from over 2.5 million official police reports. Chapters 3 and 4 describe an extensive empirical study validating the proposed framework. Chapter 3 first introduces a flat image classification architecture that extends an established SVM-based architecture using a novel feature-space local pooling algorithm. This chapter also evaluates the prediction performance of the proposed framework using models learned using the proposed architecture. Chapter 4 continues the empirical study started in the chapter 3 using deep models learned with Convolutional Neural Network-based image classification architecture. The obtained results show that flat models perform modestly compared to deep models which perform reasonably well achieving an average prediction accuracy that reaches up to 79%. This result proves our assumption that visual information contained in satellite imagery has the potential to be used as a proxy indicator of public safety. Finally, chapter 5 summarizes this study and discusses future work directions.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Ensuring public safety is an essential part of developing sustainable cities. A public safety map can assist cities to prevent future accidents, crimes, or disasters. Maps highlight patterns and trends about public safety that are difficult to spot in data collected on the ground. For example, a road-safety map made from years' worth of traffic-accident reports pinpoints roads and highways vulnerable to accidents. Similarly, a crime map highlights where within the city criminal activities abound. Such insights are invaluable in informing sustainable city-planning decision-making and policy.

However, accurate mapping requires accurate data collection, which is costly in terms of both time and money. Data collection is manual and requires skilled enumerators to conduct. While rich countries are rich in data, poor countries suffer from data poverty [1]. Therefore, city-scale public safety mapping is beyond affordable to most low- and middle-income countries. Thus, taking manual data collection out of the equation will quicken the mapping process in general, and make it possible where it is not.

Recent progress in space and imaging technologies has made satellite imagery increasingly abundant and accessible with higher resolution [2]. Satellite imagery has a bird's eye/aerial viewpoint which potentially makes it a rich medium of visual features relevant to different aspects of urban development. Given the recent breakthroughs made in the field of computer vision and pattern recognition [3], in this study we are interested in investigating predicting public safety directly from satellite imagery. In other words, investigating the use of visual information contained in satellite imagery as a proxy indicator of public safety. We present a framework for automatic city-scale public safety (road

safety and crime) mapping from raw satellite imagery using accessible tools and data sources, and aimed at developing countries.

Our motivation of predicting public safety from satellite imagery stems from the application domain we are interested in, which is predicting public safety at a city scale for the purpose of informing city-planning decision making and policy. Our motivations can be summarized as follow:

- Satellite imagery has a bird's eye/aerial viewpoint which potentially makes it a rich medium of visual features relevant to public safety. See Figures 1.1 and 1.2 for illustrated examples on the correlation between visual information in satellite imagery and road safety and crime rate respectively.

- Different from other data sources, satellite imagery has a worldwide coverage which makes it suitable for public safety prediction for almost any city around the globe.

The remainder of this chapter is organized as follows. Section 1.2 introduces the problem of public safety mapping. Section 1.3 describes contributions made in this thesis. Finally, Section 1.4 explains the organization of the thesis.

1.2 Public Safety Mapping

In this study, we define a public safety map as a city-scale visualization that describes the level of safety for a given city. We are particularly interested in road safety maps and crime maps as shown in the examples in Figures 1.3 and 1.4. Mapping previous incidents (road traffic accidents or crimes) is an established practice [4, 5] used to to gain insights on where and what interventions are needed to improve public safety. For example, a map made from manually collected reports of previous accidents visualizes where within the city road safety suffers. Maintaining and improving infrastructure around these spots helps prevent future traffic accidents. Similarly, a map of previously committed crimes highlights where within the city criminal activities abound. Increasing the frequency of police patrols around high-crime spots helps prevent future crimes. Creating a city-scale public safety map involves three main steps:

- Data collection: collecting details of previous incidents, such as location information, time and date of occurrence, category or severity level of the incident, etc.

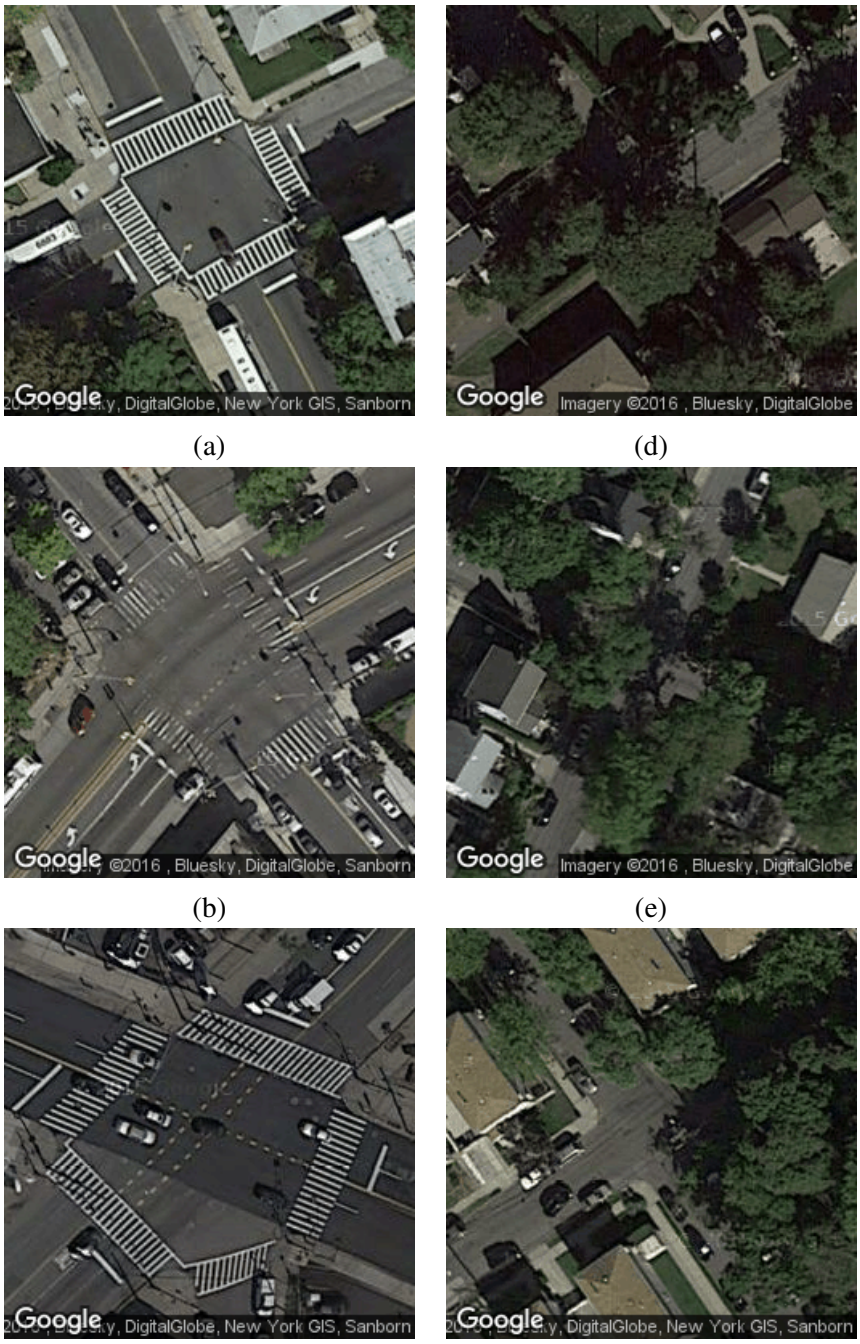


Figure 1.1: Satellite images of six different locations in New York city. Between March 2012 and March 2016, locations in the left column (a,b,c) had over 100 traffic accidents each. Those in the right column (d,e,f) had only one accident each. What is interesting is the striking visual similarity among images of the same column. Notice how images of locations of similar road safety level have similar (1) setting (highway/intersection vs. residential), (2) dominant color (gray vs. green), and (3) objects (zebra lines and vehicles vs. trees and rooftops). This example illustrates that visual features captured in satellite imagery have the potential to be used as an effective proxy indicator of road safety. Data used to create this figure can be found at: https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95

Figure 1.2: Satellite images of six different locations in the city of Chicago. Between February 2012 and January 2016, there were over 100 crimes committed in each of the locations shown in the left column (a,b,c). On the other hand and during the same period, there was only one crime committed in each of the locations of the right column (d,e,f). What is interesting is the striking visual similarity among images of the same row. Notice how images of locations of similar crime rate have similar (1) setting (highway/parking lot vs. residential), (2) dominant color (gray vs. green), and (3) objects (road lines and vehicles vs. trees and rooftops). This example illustrates that visual features captured in satellite imagery have the potential to be used as an effective proxy indicator of crime rate. Data used to create this figure can be found at: https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2/data

- Data processing: making the collected raw data more usable for later steps via conducting different operations, such as location information discretization, clustering, re-sampling, etc.
- Mapping: representing the processed data from the previous step using its location information on the city map.

Since obtaining high quality maps requires collecting data manually by skilled enumerators over long periods of time, data collection is considered as the most expensive step of the mapping pipeline. Therefore, there is a strong need for an automatic approach to public safety mapping.

1.3 Contribution of the Thesis

The major contribution of this thesis is introducing a proof-of-concept study on predicting public safety at a city scale directly from satellite imagery using tools from modern machine learning and computer vision. We summarize our contributions as follows:

- Devising an approach to obtain labeled satellite images from large-scale datasets of official police reports released as open data.
- Introducing five labeled satellite imagery datasets crawled using Google Static Maps API and mined from over 2.5 million official police reports (road accident and crime incident reports) collected by four different police departments.
- Developing a framework for automatic city-scale public safety mapping from raw satellite imagery using accessible tools and data sources aimed at developing countries.
- Proposing a novel feature-space local pooling algorithm that extends an established flat SVM-based image classification architecture.
- Providing an extensive empirical study on predicting public safety (road safety and crime rate) from raw satellite imagery using computational models learned using flat and deep image classification architectures.
- Generating several city-scale maps indicating both road safety and crime rate in three levels (low, neutral, and high) predicted directly from satellite imagery for two US cities.

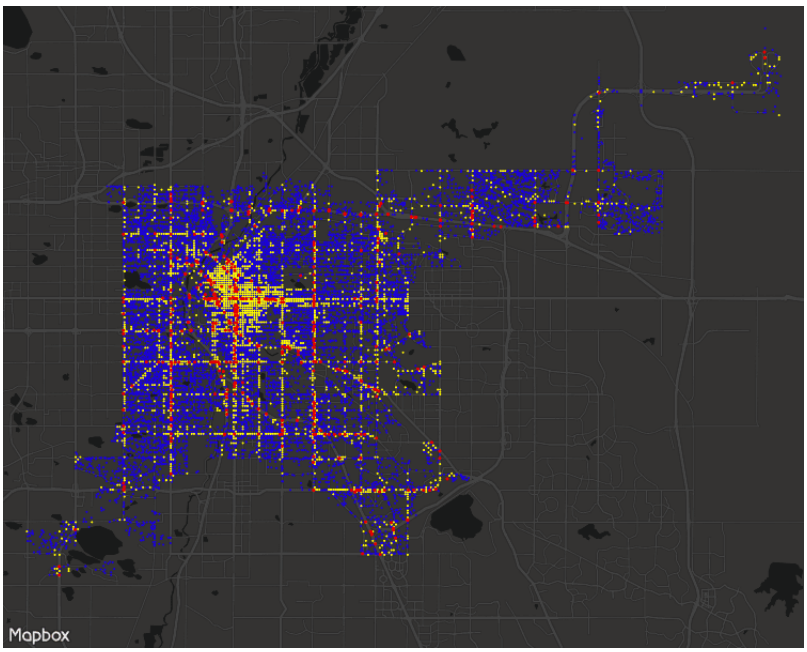


Figure 1.3: City-scale map of the city of Denver indicating road safety in three different levels: low (red), neutral (yellow), and high (blue). Data used to create this map can be found at: https://www.denvergov.org/opendata/dataset/city-and-county-of-denver-traffic-accidents

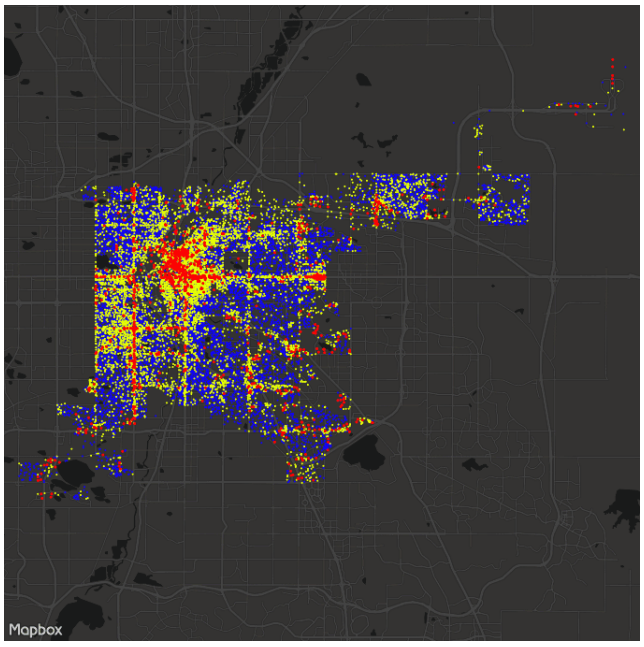


Figure 1.4: City-scale map of the city of Denver indicating crime rate in three different levels: low (red), neutral (yellow), and high (blue). Data used to create this map can be found at: https://www.denvergov.org/opendata/dataset/city-and-county-of-denver-crime

1.4 Thesis Organization

The rest of this thesis consists of four chapters. Chapter 2 overviews the contributions made in this study which can be summarized as follows: (1) proposing a framework for automatic city-scale public safety prediction from satellite imagery, (2) proposing an automatic approach for obtaining labeled satellite imagery via mining large-scale collections of official police reports released as open data, and (3) introducing five labeled satellite imagery datasets representing four different US cities, and mined from over 2.5 million official police reports. Chapters 3 and 4 describe an extensive empirical study validating the proposed framework. Chapter 3 introduces a flat image classification architecture that extends an established SVM-based architecture using a novel feature-space local pooling algorithm. This chapter also evaluates the prediction performance of the proposed framework using models learned using the proposed architecture. Chapter 4 continues the empirical study started in chapter 3 using deep models learned with a Convolutional Neural Network-based image classification architecture. The obtained results show that flat models perform poorly compared to deep models which perform reasonably well achieving an average prediction accuracy that reaches up to 79%. This result proves our assumption that visual information contained in satellite imagery has the potential to be used as a proxy indicator of public safety. Finally, chapter 5 summarizes this study and discusses future work directions.

Chapter 2

Framework for Public Safety Prediction

2.1 Introduction

In this chapter, we present the main contributions of this thesis. We start out in Section 2.2 by introducing our proposed framework for city-scale public safety prediction. Datasets of labeled satellite imagery are introduced in Section 2.3. Related works are reviewed in Section 2.4. Finally, the chapter is summarized in Section 2.5.

2.2 Proposed Framework

2.2.1 Overview

In this section, we present our proposed framework for city-scale public safety prediction using satellite imagery and open data. The assumption the proposed framework is based on is that satellite imagery is a rich medium of visual features relevant to public safety. Therefore, we propose to use visual information contained in satellite imagery as a proxy indicator of public safety. Our ultimate purpose of predicting public safety at a city scale is to automatically generate city-scale maps that indicate public safety in different levels. These maps provide insights that can be used to inform city-planning decision-making and policy.

As illustrated in Figure 2.1, the problem of public safety mapping (in the proposed framework) is formalized as a supervised image classification problem in which a city-scale satellite map is treated as a set of high-resolution satellite images each of which is assigned a safety label predicted using a computational model learned from a separate set of training samples. Given two cities, source and target cities, the goal is to generate for the target city a city-scale map indicating public safety in three

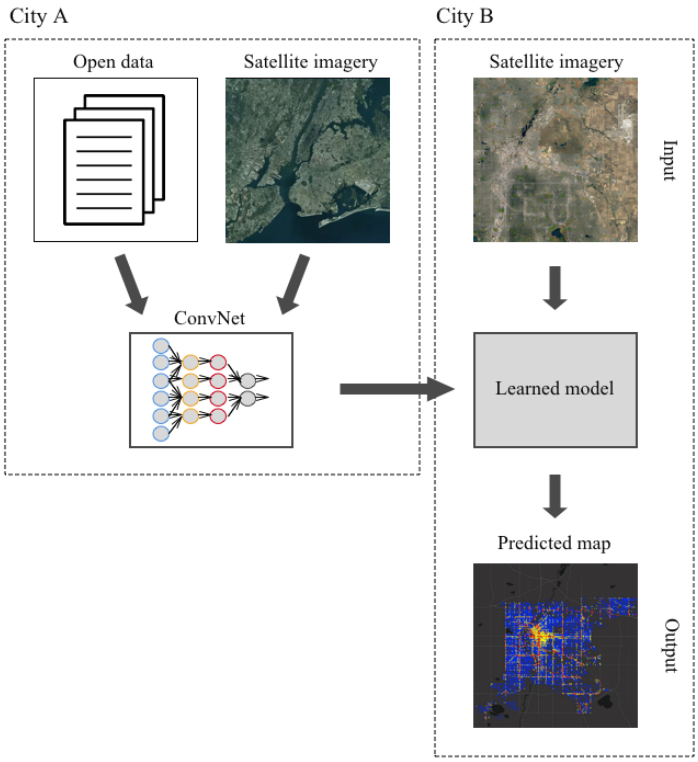


Figure 2.1: Framework for automatic public safety mapping from satellite imagery.

different levels (low, neutral, and high safety), and predicted from its raw satellite imagery.

Prediction is done using a computational model trained on data collected from the source city represented by its satellite map and official police reports and released as open data.

The proposed framework is automatic in the sense that it does not require manual data collection as in the conventional mapping pipeline explained in Chapter 1. Moreover, it makes use of previously collected data (open data) by reusing it in the form of a pre-learned knowledge (computational model). Therefore, our framework can be thought of as an automatic approach to public safety mapping suitable when proper data collection is not accessible.

2.2.2 Image Labeling

2.2.2.1 Overview

Learning a computational model able to predict public safety from raw satellite imagery first requires collecting a set of training samples labeled with public safety. To obtain our training data (labeled satellite images), we propose to mine large-scale collections of official police reports collected by police departments and released as open data.

2.2.2.2 Open Data

In this section we describe open datasets we used to obtain labeled satellite images. Open data is defined as data that can be freely used, reused and redistributed by anyone - subject only, at most, to the requirement to attribute and sharealike [6]. We used five collections of police reports released as open data by four different police departments in the US, namely New York Police Department (NYPD), Chicago Police department (CPD), Denver Police Department (DPD), and San Francisco Police Department (SFPD). These collections are organized in two categories: road accident reports, and crime incident reports. Reports follow the National Incident Based Reporting System (NIBRS) [7] in which individual incidents are described using attributes, such as time, date, geographic location, types of vehicle involved and severity level (for road accident reports), and category (for crime incident reports). Tables 2.1 and 2.2 show examples of the used reports.

We start by explaining road accident reports. We used data collected in two US cities (New York and Denver), and it is summarized as follows:

- 647,868 traffic-accident reports collected by the New York Police Department over the period between March 2012 and March 2016[1].
- 110,870 traffic-accident reports collected by the Denver city police department over the period between July 2013 and July 2016[2].

[1]https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95
[2]https://www.denvergov.org/opendata/dataset/city-and-county-of-denver-traffic-accidents

ID	Date	Time	Latitude	Longitude	Vehicle 1	Vehicle 2
1	3/12/2016	10:30	40.*******	-74.*******	Station wagon	Van
2	3/12/2016	12:15	40.*******	-74.*******	Station wagon	Unknown
3	8/31/2015	09:40	40.*******	-74.*******	Passenger vehicle	Bus
4	8/29/2015	07:08	40.*******	-74.*******	Unknown	Other
5	8/12/2014	07:31	40.*******	-74.*******	Station wagon	Bicycle
6	2/14/2016	11:34	40.*******	-74.*******	Passenger vehicle	Van
7	5/11/2016	11:14	40.*******	-74.*******	Station wagon	Unknown
8	7/29/2015	11:40	40.*******	-74.*******	Unknown	Bus
9	6/23/2015	06:18	40.*******	-74.*******	Unknown	Van
10	1/13/2014	18:39	40.*******	-74.*******	Van	Bicycle
11	3/1/2014	17:37	40.*******	-74.*******	Station wagon	Bicycle
12	12/17/2015	09:24	40.*******	-74.*******	Unknown	Van
13	5/13/2015	07:14	40.*******	-74.*******	Station wagon	Unknown
14	6/29/2014	12:43	40.*******	-74.*******	Passenger vehicle	Bus
15	4/24/2014	14:28	40.*******	-74.*******	Unknown	Van
16	1/17/2014	16:58	40.*******	-74.*******	Van	Passenger vehicle
17	11/27/2013	07:34	40.*******	-74.*******	Bicycle	Van
18	6/13/2015	06:34	40.*******	-74.*******	Van	Unknown
19	3/29/2016	17:33	40.*******	-74.*******	Unknown	Bus
20	2/14/2015	11:18	40.*******	-74.*******	Unknown	Unknown
21	11/28/2015	17:42	40.*******	-74.*******	Unknown	Station wagon
22	10/18/2014	16:37	40.*******	-74.*******	Van	Station wagon
23	7/28/2014	06:47	40.*******	-74.*******	Unknown	Passenger vehicle
24	1/29/2016	16:52	40.*******	-74.*******	Van	Station wagon
25	11/08/2013	07:22	40.*******	-74.*******	Unknown	Van

Table 2.1: Examples of NIBRS-style traffic accident reports collected by New York Police Department. Each report is described using attributes, such as date, time, location information, and types of vehicles involved in the accident. Location information is anonymized for privacy concerns.

ID	Date	Time	Latitude	Longitude	Category
1	3/18/2016	14:00	41.*********	-87.*********	Arson
2	3/18/2015	17:51	41.*********	-87.*********	Homicide
3	7/06/2013	23:00	41.*********	-87.*********	Kidnapping
4	1/14/2014	11:05	41.*********	-87.*********	Arson
5	2/24/2011	21:50	41.*********	-87.*********	Robbery
6	7/11/2013	13:00	41.*********	-87.*********	Arson
7	3/15/2013	16:57	41.*********	-87.*********	Arson
8	6/06/2013	12:00	41.*********	-87.*********	Arson
9	1/15/2015	11:05	41.*********	-87.*********	Robbery
10	5/04/2014	22:50	41.*********	-87.*********	Arson
11	8/18/2014	14:15	41.*********	-87.*********	Arson
12	6/18/2014	17:54	41.*********	-87.*********	Homicide
13	3/06/2014	15:01	41.*********	-87.*********	Arson
14	7/15/2014	13:05	41.*********	-87.*********	Robbery
15	9/04/2015	23:50	41.*********	-87.*********	Robbery
16	11/18/2015	17:00	41.*********	-87.*********	Arson
17	12/18/2015	17:41	41.*********	-87.*********	Robbery
18	7/06/2013	15:00	41.*********	-87.*********	Kidnapping
19	6/15/2015	11:05	41.*********	-87.*********	Robbery
20	6/04/2015	16:50	41.*********	-87.*********	Robbery
21	5/18/2015	12:00	41.*********	-87.*********	Arson
22	9/18/2015	15:51	41.*********	-87.*********	Homicide
23	4/06/2013	17:00	41.*********	-87.*********	Kidnapping
24	2/15/2013	19:05	41.*********	-87.*********	Robbery
25	2/04/2013	22:50	41.*********	-87.*********	Arson

Table 2.2: Examples of NIBRS-style crime-incident reports collected by the Chicago Police Department. Each report is described using attributes, such as date, time, location information, and category of the incident. Location information is anonymized for privacy concerns.

Category	City	Source	No. of reports
Road safety	New York	NYPD	647,868
Road safety	Denver	DPD	110,870
Crime	Chicago	CPD	1,028,885
Crime	Denver	DPD	198,506
Crime	San Francisco	SFPD	652,807

Table 2.3: Summary of the used police report datasets. We have used five different datasets of police reports openly released by New York police department, Chicago police department, Denver police department and San Francisco police department. In total we used over 2.5 million police reports categorized in two different categories: road safety and crime.

As for crime reports we used data collected in three US cities (Chicago, Denver, and San Francisco), and its summarized as follows:

- 1,028,885 crime reports collected by the Chicago Police Department over the period between September 2001 and August 2016[3].
- 198,506 crime reports collected by the Denver city police department over the period between July 2014 and July 2016 [4].
- 652,807 crime reports collected by the San Francisco Police Department over the period between March 2003 and September 2016[5].

See Table 2.3 for a summary of all open datasets we used in this study. The procedure for mining labeled satellite images from police reports is explained next.

2.2.2.3 Procedure

The following steps explain the procedure we followed to obtain labeled satellite images from police reports:

Location information discretization

Using a square grid, we divided the input city-scale satellite map into square regions (r). Then given their location information, incidents (accidents or crimes) documented by the corresponding police

[3]https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2
[4]https://www.denvergov.org/opendata/dataset/city-and-county-of-denver-crime
[5]https://data.sfgov.org/Public-Safety/SFPD-Incidents-from-1-January-2003/tmnf-yvry

departments were assigned to different regions. Finally, each region is assigned a safety score (S_r) given as the sum of all accidents/crimes occurred within its boundaries during the studied period:

$$S_r = \sum_{i=1}^{n} a_{i,r}, \tag{2.1}$$

where $a_{i,r}$ is the i-th incident occurred within the boundary of region r, and n is the total number of incidents.

Binning

In order to obtain three safety labels (low, neutral, and high), we clustered the obtained safety scores (from the previous step) by frequency around three bins using the k-means algorithm [8], such that:

$$\arg\min_{T} \sum_{i=1}^{k} \sum_{x \in T_i} \|x - \mu_i\|^2, \tag{2.2}$$

where μ_i is the mean of the points in T_i, $k = 3$ is the number of bins, and x is the frequency of individual scores. We have experimented with other clustering algorithms, such as Gaussian Mixture Models (GMM) and Jenks natural breaks optimization [9]. However, we found that k-means gives the best results.

Resampling

Given that the obtained three classes are highly imbalanced and in order to avoid learning a biased model, we resampled our data via downsampling majority classes so that the three classes are balanced out.

Finally, we represented each of the regions with a satellite image centered around the location information (GPS coordinates) of its center. These images are to be used later to train, verify, and test our learned models.

2.3 Labeled Satellite Imagery

Following the procedure explained in the previous section, we mined the previously introduced open datasets and obtained five datasets of satellite images labeled with public safety. The obtained datasets represent four different US cities and are organized in two different categories: road safety and crime. See Figure 2.2 for a sample of the collected images. The obtained datasets are described in the following (See Table 2.4 for a summary):

Category	Name	No. of reports	Size	Labels
Road safety	*New York*	647,868	14,000	Low, neutral, high
Road safety	*Denver1*	110,870	21,406	Low, neutral, high
Crime	*Chicago*	1,028,885	12,000	Low, neutral, high
Crime	*Denver2*	198,506	25,169	Low, neutral, high
Crime	*San Francisco*	652,807	19,897	Low, neutral, high

Table 2.4: Satellite imagery datasets mined from over 2.5 million official police reports. In total we have collected five datasets spanning four different US cities. Datasets are organized in two different categories: road safety and crime. Individual images are labeled with one of three safety labels: low, neutral, and high safety.

2.3.1 Road Safety

- *New York*: 14,000 satellite images obtained from official traffic-accident reports collected by the New York Police Department (NYPD).
- *Denver 1*: 21,406 satellite images obtained from official traffic-accident reports collected by the Denver city Police Department.

2.3.2 Crime

- *Chicago*: 12,000 satellite images obtained from official crime reports collected by the Chicago Police Department.
- *Denver 2*: 25,169 satellite images obtained from official crime reports collected by the Denver city Police Department.
- *San Francisco*: 19,897 satellite images obtained from official crime reports collected by the San Francisco Police Department.

2.4 Related Works

In this section, we review previous works on city-scale public safety mapping using machine learning and compare them to ours. We first start with works on road safety mapping in Section 2.3.1. Then, in Section 2.3.2 we cover works on urban safety (crime) mapping.

Figure 2.2: Examples of the collected satellite images. Upper rows (a-f) are random road safety samples. Bottom rows (g-l) are random urban safety (crime) samples. Images are individually labeled with one of three safety labels (from left to right: low, neutral, and high safety).

2.4.1 Road Safety

To the best of our knowledge, [10] is the only work that uses machine learning to predict city-scale road safety maps. In this work, a computational model is learned from traffic-accident reports and human mobility data (i.e., GPS data) collected from 1.6 million smartphone users over a period of seven months. The learned model is then used to predict from real-time GPS data a map for the city of Tokyo indicating road safety in three different levels.

This works is similar to ours in the fact that it uses patterns recognized in an abundant and unstructured source of data as a proxy indicator of road safety. While Chen et al. use real-time GPS data, we use satellite imagery as our abundant source of data. However, the core difference between the two works is the application domain each is intended for. While Chen et al. are interested in generating user-oriented maps intended for real-time use, we are interested in generating maps for the purpose of informing city-planning decision-making and policy and eventually improve road safety for cities where proper data collection is not accessible.

It is worth mentioning that for the application we are interested in, using satellite imagery rather than GPS data is more practical since:

- Satellite images are ubiquitous (Available for free on Google Maps, for instance).
- Smartphones in low- and middle-income countries (which this research is targeting) are not as widely used as in high-income countries, i.e., GPS data in developing countries can not be used as a reliable indicator of road safety at a city scale.

We are aware of other works, such as [11–14], which mainly focus on the detection and analysis of traffic accident-prone areas (also known as, traffic accident hotspots) rather than the prediction of road safety at a city scale. Therefore, and given the above, we believe that our work is the first to attempt using machine learning to predict city-scale road safety maps directly from raw satellite imagery.

2.4.2 Urban Safety (Crime)

To the best of our knowledge, the first major effort made at predicting city-scale urban safety maps is described in [15]. First, using an online crowdsourcing platform, a group of 7872 participants were shown random pairs of 4019 Google Street View images collected from the cities of New York,

Boston, Salzburg, and Linz. For each pair, the participants were asked to choose the image they think looks safer. Then, individual images were assigned safety scores obtained from the accumulated preference vectors using the TrueSkill algorithm [16]. Finally, each image was represented with a set of generic visual features collectively used to learn a computational model. The learned model was later used to generate city-scale safety maps for 27 other US cities predicted directly from their Google Street View images. This study was recently extended in [17] to cover 29 more cities, using models learned from much larger pool of images annotated by over 81,000 participants.

Our work is similar to [15, 17] in that both use visual information as a proxy indicator of urban safety. While [15, 17] use Google Street View Images, we use satellite imagery instead.

On the other hand, the core difference between the two lies in the definition of urban safety. While in [15, 17], urban safety is subjectively judged by participants, we define urban safety based on the rate of crimes committed as reported by police departments.

Compared to ours, we believe that the mapping approach reported in [15, 17] has the following limitations:

- It is only viable in cities that have services similar to Google Street View available. It cannot be applied in most cities of low- and middle-income countries.
- Building a robust model that can predict urban safety from natural images requires crowdsourcing the votes of tens of thousands of online participants, a process that is both time consuming and labor intensive.

We are aware of other works, such as [18], which mainly focus on the prediction of crime-prone areas (crime hotspots) rather than the prediction of crime at a city scale. Therefore, and given the above, we believe that our work is the first to attempt using machine learning to predict city-scale crime maps directly from raw satellite imagery.

2.5 Summary

In this chapter, we introduced our proposed framework for public safety prediction in Section 2.2. In the same section we also explained our approach to obtain labeled satellite images from police reports released as open data. In Section 2.3, we introduced five datasets of labeled satellite images mined

from over 2.5 million official police reports to be used later to train, verify and test our models. We finally reviewed previous works on machine learning-based city-scale public safety prediction/mapping in Section 2.4.

In the following two chapters, we present the results of an extensive empirical study we have conducted to validate the effectiveness of the proposed framework.

Chapter 3

Prediction Using Flat Models

3.1 Introduction

In this chapter, we evaluate the performance of the proposed framework using computational models learned using a flat image classification architecture. Performance is evaluated for two tasks: road safety and crime rate prediction tasks. The remainder of this chapter is organized as follows. The used flat classification architecture is presented in Section 3.2. Our proposed pooling extension is described in Section 3.3. Empirical results are given in Section 3.4. Finally, the chapter is summarized in Section 3.5.

3.2 Flat Image Classification Architecture

3.2.1 Background

At the heart of modern image recognition lies a local patch-based multi-layer architecture that has significantly evolved during the past decade. This architecture can be summarized as follows. First, handcrafted descriptors (e.g., SIFT [19], HOG [20], SURF [21], etc.) densely sampled from an input image are projected into a codebook space using a common coding method, such as vector quantization (*coding step*). Second, a fixed-length, global image representation is generated via summarizing the encoded descriptors, obtained in the previous step, over the image's area (*pooling step*). In the classification task, this pooled representation is finally fed to a linear (or nonlinear) classifier where both training and class label prediction take place. Extensions to this architecture (e.g., [22–24]) have dominated standard classification benchmarks (e.g., Pascal VOC [25]) for several years. As mentioned above, this architecture has been refined greatly with improvements aimed at both of its steps.

In this chapter, we propose a novel extension to this architecture that improves its pooling step.

The idea of pooling originates in the Nobel-winning work of Hubel and Wiesel on the mammalian visual cortex [26] in which they explain a cascaded model of the visual cortex where responses coming from lower simple cells are aggregated before being fed to higher complex cells, rendering them invariant to small spatial transformations. This seminal work has long inspired computer vision researchers to adopt the idea of pooling for the aim of building robust translation-invariant visual recognition systems. Thus, pooling has been a genuine component in visual recognition all the way from the early Neocognitron [27], to the Bag-of-Words (BoW) model [28, 29], up until the recently rediscovered convolutional neural networks [30]. In its most basic adaptation, pooling summarizes the image's features by taking the average (or max) value of their activations [31].

Pooling involves two components: (1) an operator and (2) a neighborhood. While the operator does the summarization function, the neighborhood determines which descriptors are to be pooled together. In conventional pooling (e.g., [28, 29]), the pooling operator is applied to all encoded descriptors of the input image at once, i.e., the pooling neighborhood is defined as the whole area of the image. While the direct advantage of this pooling is added robustness to input translations, its major disadvantage is inevitable information loss. To compensate for part of this loss, an extension to pooling (local pooling) enforces locality via jointly pooling only descriptors that are members of a certain local neighborhood. A local neighborhood could be any subgroup of the image's descriptors that are "close" according to a certain criterion. Based on the space within which local neighborhoods are defined, work on local pooling can be categorized into: (1) image-space and (2) feature-space methods. A local neighborhood in the image space could be a subregion (object) within the image plane. On the other hand, a local neighborhood in the feature space could be a partition (bin) whose members share some aspect in common (e.g., visual similarity). As it might be more straightforward to pool descriptors based on their spatial location within the image, the bulk of the work on local pooling has focused on the image space [22, 32, 33]. However, our method operates in the feature space as we believe in the highly untapped potential this space holds.

Within the adopted pipeline (reviewed in the following), the most notable work on local pooling in the feature space seems to be [34], in which, in the same spirit as that of [35, 36], the image representation is generated via (1) clustering the extracted descriptors over a handful of codewords of

a universal codebook learned via k-means clustering and (2) applying the pooling operator within each obtained cluster individually. The final image representation is the (normalized) concatenation of the pooled features. Partitioning of the input data by minimal Euclidean distance (i.e., clustering) assures that only visually similar descriptors are pooled together. In other words, the notion of closeness in the feature space is defined in terms of the visual appearance of descriptors. This method is simple and can be regarded as a straightforward extension to the popular spatial pyramid (SP) model [22] within the feature space.

In this work, we mainly try to determine whether partitioning the feature space using a k-means codebook, i.e., based on visual appearance only as in [34–36], is optimal for local pooling in the image classification task. While k-means clustering preserves, to some extent, the visual similarity between descriptors, it totally discards any class-related information (i.e., high-level semantics) of the input image. For example, two visually similar descriptors belonging to two semantically different objects (subregions) within the image will be assigned to the same pooling bin. In this case, jointly pooling the two descriptors totally discards the image's semantics.

Motivated by the above observation, we aim at generating pooling bins that are aware of the semantics of the input image. To this end, we propose partitioning the feature space over clusters of visual prototypes common to images belonging to the same category (i.e., semantically similar images). The clusters in turn are generated via simultaneously clustering (co-clustering) images and their visual prototypes (codewords). The co-clustering is applied offline on a subset of training data and conducted using Bregman co-clustering [37]. Therefore, contrary to features pooled from appearance-based partitioning [34–36], our features are aware of the semantic context of the input image within the dataset, which consequently boosts classification performance. Similar to [34], spatial information can be easily encapsulated via implementing our local pooling within an SP or any other similar method.

3.2.2 Classification Pipeline

We are interested in the coding-pooling pipeline of image classification [38]. This pipeline is summarized in four successive steps: (1) feature extraction, (2) coding, (3) pooling, and finally (4) classification. Individual steps are explained below.

Feature extraction

Given an input image $I \in \mathbf{I}$ (the image dataset), a set of low-level features (e.g., SIFT) sampled at N different locations is extracted, such that $\mathbf{X} = \{x_i\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^d$ is the d-dimensional low-level feature extracted at location i. Several methods have been proposed in the literature to obtain salient regions within the image from which features are extracted (See [39] for a detailed comparison). However, in the classification task, it has been shown in [22] that better performance is obtained when features are densely sampled from a regular grid covering the image plane.

Coding

The first step is to train a codebook $\mathbf{B} = [b_1, \cdots, b_K] \in \mathbb{R}^{d \times K}$, where $\{b_i\}_{i=1}^{K}$ is the set of the d-dimensional codewords obtained via unsupervised learning, such as k-means clustering. Note that individual codewords belong to the same space to which the extracted features, of the previous stage, belong. Then, given a coding function ψ, the extracted features ($\mathbf{X}$) of the input image are individually projected into the space of the learned codebook. More formally, each descriptor $x_i \in \mathbb{R}^d$ is mapped to a new representation $v_i \in \mathbb{R}^K$, using a coding function $\psi : \mathbb{R}^d \rightarrow \mathbb{R}^K$, such that:

$$v_i = \psi(x_i), \quad \forall\ i \in \{1, \cdots, N\}. \tag{3.1}$$

The coding function can be thought of as an activation function for the codebook, activating each of the codewords according to the input descriptor [40]. Depending on the coding function used, activations are either continuous or binary-valued. A multitude of coding functions (algorithms) have been proposed in the literature. In the following, we explain three of the most popular ones: Vector Quantization (VQ), Sparse Coding (SC) [23], and Locality-constrained Linear Coding (LLC) [24]. See [41] for a comprehensive survey on coding functions.

Vector Quantization (VQ) encodes each descriptor by assigning the value 1 to its closest codeword and zeros to the rest. This is done via solving the following constrained least squares fitting problem:

$$\begin{aligned} \arg\min_{\mathbf{V}} \quad & \sum_{i=1}^{N} \|x_i - \mathbf{B}v_i\|^2 \\ \text{subject to} \quad & \|v_i\|_{\ell_0} = 1, \quad \text{and} \quad \|v_i\|_{\ell_1} = 1, \quad v_i \geq 0 \end{aligned} \tag{3.2}$$

where $\mathbf{V} = [v_1, v_2, \cdots, v_N] \in \mathbb{R}^{K \times N}$ is the matrix of codes obtained for the set $\mathbf{X}$. With a single non-

zero element (i.e., $\|v_i\|_{\ell_0} = 1$), these codes are highly sparse. This leads to a high quantization loss, especially when the descriptor being encoded is close to several codewords at the same time.

To alleviate the quantization loss of VQ, Sparse Coding (SC) approximates each descriptor as a sparse linear combination of the codewords. In other words, SC relaxes the cardinality constraint ($\|v_i\|_{\ell_0} = 1$) in Eq. (3.2). This is achieved via solving the following optimization:

$$\arg\min_{\mathbf{V}} \sum_{i=1}^{N} \|x_i - \mathbf{B}v_i\|^2 + \lambda\|v_i\|_{\ell_1}, \tag{3.3}$$

where λ is a parameter that controls the sparsity of the obtained code induced by the ℓ_1 norm.

Finally, approximate Locality-constrained Linear Coding (LLC) addresses the non-locality that can occur in SC via encoding each descriptor with its n-nearest codewords. In other words, a new codebook $\mathbf{B}(x_i, n)$ is constructed for each descriptor x_i, such that $\mathbf{B}(x_i, n) = NN_n(x_i, \mathbf{B}) \in \mathbb{R}^{d\times n}$, where n ($n \ll K$) is a constant that defines how localized the coding is. Approximate LLC is formulated as:

$$\begin{aligned} \arg\min_{\mathbf{V}^*} \quad & \sum_{i=1}^{N} \|x_i - \mathbf{B}(x_i, n)v_i^*\|^2 \\ \text{subject to} \quad & 1^T v_i^* = 1, \end{aligned} \tag{3.4}$$

where $v_i^* \in \mathbb{R}^n$ is the obtained n-dimensional code, later projected into the original space ($\mathbb{R}^K$) of the learned codebook.

Pooling

At this stage, the matrix $\mathbf{V} \in \mathbb{R}^{K\times N}$ of encoded descriptors is transformed into a fixed-length global image representation $z \in \mathbb{R}^K$. This is achieved via applying the pooling operator $\phi : \mathbb{R}^{1\times N} \to \mathbb{R}$ to each row of $\mathbf{V}$ separately. The final image representation is the concatenation of the pooled K descriptors, such that:

$$z = [z_1, z_2, \cdots, z_K]^T, \tag{3.5}$$

where $z_k \in \mathbb{R}$ is given:

$$z_k = \phi\Big(\{v_{ki}\}_{i=1}^{N}\Big), \quad \forall\, k \in \{1, \cdots, K\}, \tag{3.6}$$

where v_{ki} is the activation value of the i-th descriptor to the k-th codeword. Several pooling operators have been proposed in the literature. The reader is referred to [42] for a recently published survey on the topic.

Classification

Both training and class label prediction take place at this stage. The pooled image feature $z \in \mathbb{R}^K$ is (normalized and then) fed to a classifier. A standard classifier choice is Support Vector Machines (SVM) [43].

3.3 Proposed Pooling Extension

In this section, we describe our proposed pooling extension. We start out by detailing how the feature space is partitioned. Then, we explain how the final image representation is generated. Finally, we compare our method to related works.

3.3.1 Feature-space partitioning

To obtain pooling bins, we need to partition the feature space. This section details this procedure.

3.3.1.1 Introduction

Given an image's extracted low-level features $\mathbf{X}$, our goal is to find P different neighborhoods $\{x_i\}_{i=1}^{N_p}$, $\forall\ p \in \{1, \cdots, P\}$, within $\mathbf{X}$, so that members of each neighborhood are semantically coherent. In this work, semantics are defined as the high-level visual traits common to images conveying the same concept, i.e., belonging to the same category, and by "high-level" we mean characteristics that go beyond the exact appearance of individual images and ascribe to their semantic context within the dataset. Therefore, favoring simplicity, we propose to model semantics as *clusters of visual prototypes* (codewords) common to images belonging to the same category.

To this end, we make use of an established data mining tool called co-clustering [44]. A co-clustering algorithm simultaneously clusters rows and columns of an input data matrix and produces two correlated sets of clusters representing the two dimensions of the input (rows and columns) as an output. Thus, as shown in [45, 46], semantics of a given dataset can be captured, in the form of clusters of visual prototypes, by co-clustering a subset of the dataset's training images represented as a matrix of Bags of Words (BoWs).

To conduct the co-clustering, we use [37] in which optimal co-clustering is guided by a search for the nearest matrix approximation that has the minimum Bregman information. Before explaining the co-clustering procedure, in the following we introduce two preliminary concepts: Bregman divergence and Bregman information.

3.3.1.2 Bregman divergences and Bregman information

First introduced in [47], Bregman divergences define a large class of widely used loss functions, such as the squared Euclidean distance, KL divergence, etc. Given a convex function f, the Bregman divergence between two data points $a_1, a_2 \in \mathbb{R}$ is defined as:

$$d_f(a_1, a_2) = f(a_1) - f(a_2) - < \nabla f(a_2),\ a_1 - a_2 >, \tag{3.7}$$

where $< a_1, a_2 >$ is the inner product between a_1 and a_2, and ∇ is the gradient operator. The convexity of f guarantees that $d_f(a_1, a_2)$ is non-negative for all $a_1, a_2 \in \mathbb{R}$. By choosing a suitable convex function (f), the Bregman divergence can generalize several existing distance measures. For instance, using the convex function $f(a) = a \log a$ defined over $a \in \mathbb{R}$, the KL divergence between two points $a_1, a_2 \in \mathbb{R}$ (i.e., $D_{KL}(a_1 \parallel a_2)$) can be expressed as a Bregman divergence as:

$$d_f(a_1, a_2) = a_1 \log(a_1/a_2) - (a_1 - a_2). \tag{3.8}$$

Based on Bregman divergences, we explain another concept called Bregman information [37]. Given a Bregman divergence (d_f) and a random variable (A), the uncertainty of A can be captured in terms of a useful concept called Bregman information (I_f), defined as the expected (E) Bregman divergence to the expectation, such that:

$$I_f(A) = E[d_f(A,\ E(A))]. \tag{3.9}$$

In the following, we explain Bregman co-clustering in which optimal co-clustering is guided by a search for the nearest (in Bregman divergence) approximation matrix that has the minimum Bergman information.

3.3.1.3 Co-clustering images and visual prototypes

Consider a subset of j training images $C = \{c_v\}_{v=1}^{j}$, spanning L different categories, represented as BoWs generated by using a codebook of m visual prototypes $R = \{r_u\}_{u=1}^{m}$. These images can be regarded as a data matrix $A \in \mathbb{R}^{m \times j}$ of two underlying discrete random variables R and C representing rows (visual prototypes) and columns (images), respectively. The aim here is to simultaneously cluster columns (C) into L categories $\hat{C} = \{\hat{c}_h\}_{h=1}^{L}$ and rows (R) into P clusters $\hat{R} = \{\hat{r}_g\}_{g=1}^{P}$. The obtained co-clustering can be thought of as a pair of mapping functions $\hat{R} = \rho(R)$ and $\hat{C} = \gamma(C)$ operating on the rows and columns, respectively.

According to Bregman co-clustering [37], the optimal solution is the pair (ρ, γ) that constructs the nearest approximation matrix that has the minimum Bregman information, i.e., satisfying:

$$\arg\min_{(\rho,\gamma)} \; E[d_f(A,\; \hat{A})], \tag{3.10}$$

where $\hat{A}$ is the approximation matrix with the minimum Bregman information among the set of approximations that satisfy Eq. (3.10). Based on the nature of the input data, different Bregman divergences can be used to run the co-clustering. However, it has been shown in [37] that KL divergence is best suited as a loss function when the input matrix (A) is the joint probability distribution ($p(R, C)$) of the underlying discrete random variables. Thus, as explained previously, by using a suitable convex function, KL divergence can be expressed as a Bregman divergence as in Eq. (3.8). This in turn means that Bregman co-clustering reduces to the information-theoretic co-clustering of [48] in which the optimal co-clustering is the one that minimizes the following:

$$\begin{aligned} \Delta MI \quad &= MI(R; C) - MI(\hat{R}; \hat{C}) \\ &= D_{KL}\Big(p(R, C) \parallel q(R, C)\Big), \end{aligned} \tag{3.11}$$

where $MI(R; C)$ is the mutual information between two discrete random variables R and C and is given as:

$$MI(R; C) = \sum_{r \in R, c \in C} p(r, c) \log\Big(\frac{p(r, c)}{p(r)p(c)}\Big), \tag{3.12}$$

and $q(R, C)$ is a distribution of the form:

$$q(R, C) = p(\hat{R}, \hat{C})\ p(R|\hat{C})\ p(C|\hat{C}). \tag{3.13}$$

Therefore, optimal co-clustering can be obtained by searching for the nearest approximation matrix that has a distribution of the form shown in Eq. (3.12). To this end, [48] proposed a neat algorithm that is computationally efficient even for sparse data (our case). As an input, the algorithm takes the joint probability distribution function $p(R, C)$, the number of categories (L), and the number of row clusters (P). As an output, the algorithm produces the pair (ρ, γ). The algorithm starts (at $t = 0$) with a random pair (ρ^t, γ^t) which is updated at each iteration (t) via: (1) clustering the rows (R) while keeping the columns (C) fixed and (2) clustering the columns while keeping the rows fixed. The algorithm stops when Eq. (3.11) is less than a preset threshold.

3.3.2 Image representation

Now we explain how the final image representation is generated. Given an input image $I \in \mathbf{I}$, its set of extracted low-level features ($\mathbf{X}$) are first clustered over the (row) clusters ($\hat{R} = \{\hat{r_g}\}_{g=1}^{P}$) learned via co-clustering training images and their visual prototypes into P different neighborhoods. Then, by using a k-means codebook, each neighborhood is individually pooled into a K-dimensional feature vector ($z_p \in \mathbb{R}^K$), such that:

$$\begin{aligned} z_p &= [z_{p1}, z_{p2}, \cdots, z_{pK}]^T, \\ \text{where} \quad z_{pk} &= \phi\Big(\{v_{ki}\}_{i=1}^{N_p}\Big). \end{aligned} \tag{3.14}$$

The final image representation (z)[1] is then the concatenation of the P individually pooled features (z_p):

$$z = [z_1^T, z_2^T, \cdots, z_P^T]^T \in \mathbb{R}^{PK}. \tag{3.15}$$

Similar to [34], spatial information can be easily encapsulated in the image representation by repeatedly pooling features locally within the individual spatial cells of an SP.

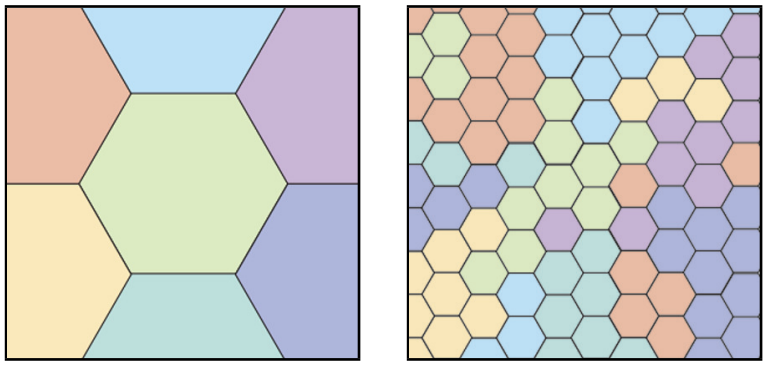
3.3.3 Semantically enhanced pooling bins

Here we discuss the nature of the feature-space partitioning (pooling bins) obtained in our method and how it compares to the appearance-based partitioning of [34–36]. As previously explained, the

[1] This representation (z) along with the image's label are what passed to the SVM classifier later.

feature space in our method is partitioned by clustering the input image's extracted descriptors ($\mathbf{X}$) over clusters of visual prototypes ($\hat{R}$) learned through Bregman co-clustering. However, given the fact that the co-clustering operates on the training BoWs generated using an m-dimensional k-means codebook ($R = \{r_u\}_{u=1}^{m}$), we can say that our partitioning can be regarded as obtained in two successive steps: (1) clustering over m ($m \gg P$) k-means codewords followed by (2) aggregating the m clusters of the previous step into P bins using a map ($\hat{R} = \rho(R)$) learned via Bregman co-clustering. Given that the learned map captures the semantic context of the dataset at hand [45], our pooling bins can be regarded as being semantically enhanced compared to those learned in [34–36], in which the image's descriptors are directly clustered over P codewords of a k-means codebook.

Figure 3.1 illustrates a cartoon representation of an appearance-based partitioning compared to a semantically enhanced one (ours). Notice that (1) both spaces have the same number of pooling bins (number of unique colors), i.e., the pooled image representation has exactly the same dimension in both spaces., and (2) our bins are disjoint in the feature space.



(a) (b)

Figure 3.1: Cartoon representation of (a) an appearance-based partitioning compared to (b) ours. Different colors represent different pooling bins. Number of pooling bins is the same in both spaces. Contrary to (a), our bins (b) are disjoint in the feature space. Our partitioning can be seen as obtained via (1) clustering the input over a large k-means codebook and then (2) aggregating semantically coherent bins according to the result of the co-clustering.

3.4 Experimental Results

In this section, we present the results of two separate experiments. In Section 3.4.1, we present the results of empirically validating the proposed pooling extension and compare it to related works. In Section 3.4.2, we present the results of an empirical study we conducted to evaluate the performance of the proposed framework using models learned as detailed in Sections 3.2 and 3.3.

3.4.1 Experiment (1)

3.4.1.1 Experimental protocol

Our experimental protocol is explained here. An overview of the used image datasets is given first, followed by an explanation of the implementation details.

Image datasets

In our experiments, we used *Caltech-101*, *Caltech-256*, *15 Scenes*, and *17 Flowers* image datasets. Individual datasets are briefly introduced in the following:

- *Caltech-101* [49]: This is a widely used dataset suitable for the generic-object classification task. It consists of 9144 images exhibiting a variety of objects spanning 102 different categories (e.g., *person*, *cougar*, etc.). The number of images per category ranges from 31 to 800. Images come in an approximate resolution of 200×300 pixels each.

- *Caltech-256* [50]: This is a challenging generic-object classification dataset that consists of 30607 images organized in 257 categories of the same nature as those of *Caltech-101*. The number of images per category is 80 to 827. Images come in an approximate resolution of 200×300 pixels each.

- *15 Scenes* [22, 51, 52]: This is a common choice for the task of scene classification, and the dataset consists of 4485 images organized in 15 different categories of indoor (e.g., *kitchen*, *bedroom*, etc.) and outdoor (e.g., *forest*, *highway*, etc.) scenes. Each category has 200 to 400 images on average. Images come in an average size of 250×300 pixels each.

- *17 Flowers* [53]: This is a dataset of 1360 high-resolution flower images organized in 17 different categories. Each category has 80 images. Images have large scale, pose and light variations. *17 Flowers* is a challenging fine-grained classification dataset.

Implementation details

Favoring the reproducibility of our results, the implementation details of our experiments are explained in this section.

- *Pre-processing*: Images were first converted to grayscale and then reduced in resolution so that the longest side was less than or equal to 300 pixels.

- *Feature extraction & description*: Using VLFeat toolbox [54], low-level features were densely sampled over a rectangular grid of 16×16 pixel patches with a sampling rate of 4 pixels. Unless otherwise noted, a 128-dim SIFT descriptor was then computed for each extracted patch.

- *Codebooks*: Standard k-means clustering was used to generate codebooks. The number of codewords was always set to 4096.

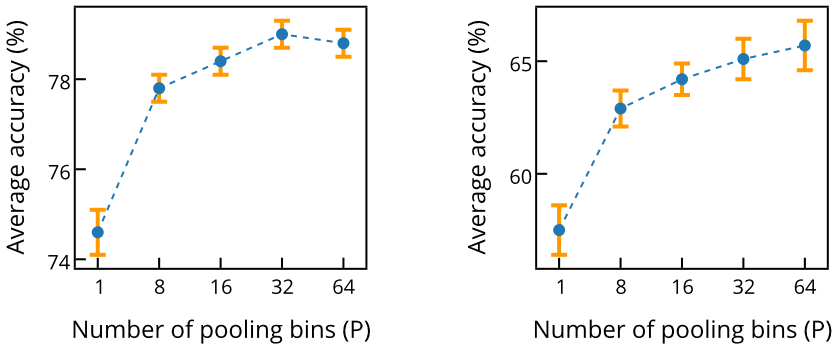
- *Coding, pooling (operator), and normalization*: Unless otherwise noted, the combination of sparse coding and max pooling was used in our experiments. The final image representation is always ℓ_2-normalized.

- *Co-clustering*: We applied Bregman co-clustering offline on the training data of each dataset for a number of row clusters $P = \{8, 16, 32, 64\}$.

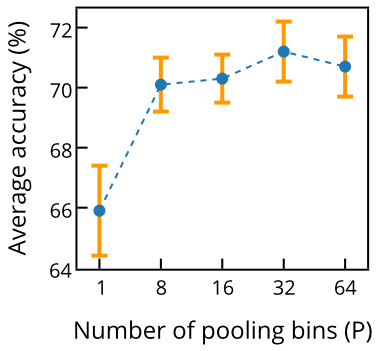
- *Spatial information*: We used a three-layer spatial pyramid of 21 cells (1×1, 2×2, 4×4) whenever spatial information was included. Similar to [34], our local pooling is easily implemented within an SP via repeatedly pooling features locally within its individual spatial cells. The final image representation is the concatenation of the locally pooled features across all cells. This representation is finally fed to a classifier.

- *Classification*: We adopted the one-versus-all methodology by training one SVM classifier per class using the library reported in [55]. The cost parameter was determined by cross-validation within the training data of the target dataset. Following the common practice of training/testing, we used 30 training images per class for *Caltech-101*, 60 for *Caltech-256*, 100 for *15 Scenes*, and 40 for *17 Flowers*. The rest were used for testing.

- *Evaluation*: Average classification accuracy and standard deviation, over s runs, are reported as classification results. The number of runs (s) is set to 10 for all datasets except for *17 Flowers*, where training/testing data splits are provided by the authors.



(a) *Caltech-101* (b) *15 Scenes*



(c) *17 Flowers*

Figure 3.2: Classification accuracy (%) comparison among the method in previous work (blue), our method (green) and random (orange). On (a) and (b), features pooled from an appearance-based bins (previous work) almost always perform worse than those pooled from random bins.

3.4.1.2 Results (1)

We empirically analyze the performance of the proposed method within the feature space only. In other words, spatial information is not included at all here (i.e., our method is not implemented within an SP). Thus, results reported here are by no means intended to be compared with the published state-of-the-art methods. For such a comparison, please refer to the following subsection which is dedicated to this purpose. This style of reporting experimental results has been previously adopted

by others including [56] and [57]. We start by assessing the performance improvement our method brings to the baseline. Then, we compare our method to a closely related work on local pooling in the feature space.

Contribution to the baseline

The purpose of this study was to empirically assess the performance improvement our method brings to the baseline, i.e., how locally pooling image features from a space partitioning obtained by Bregman co-clustering boosts the classification performance of the baseline. As a classification baseline, we adopted the Bag-of-Features (BoW) model, implemented as previously detailed. We chose to analyze the contribution of our method in generic-object, scene, and fine-grained classification scenarios. Thus, experiments were conducted on *Caltech-101*, *15 Scenes*, and *17 Flowers* image datasets.

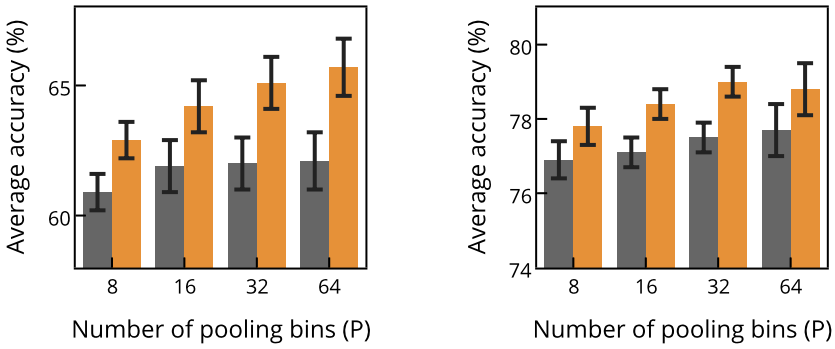
Figure 3.2 compares classification performance of the baseline ($P = 1$)[2] to that of our method implemented for an increasing number of pooling bins $P \in \{8, 16, 32, 64\}$. From the results, it is clear that local pooling in the feature space always improves classification performance over the baseline for all datasets. This was observed in a previous work [56]. Moreover, doubling the number of pooling bins always boosts performance on the first dataset. However, for both the second and third datasets, performance degrades when 64 pooling bins are used. In summary, performance boost ranges between 5.4% and 8.2% for *Caltech-101*, 3.2% and 4.4% for *15 Scenes*, and 4.2% and 5.3% for *17 Flowers*.

To confirm that our implementation of the baseline achieves results comparable to the recently published results, we implemented the baseline within a spatial pyramid. We obtained 76.8 ± 0.8 and 82.7 ± 0.3 on *Caltech-101* and *15 Scenes*, respectively. These results are very close to (slightly better than) those in [58] in which similar experimental settings were followed. As for *17 Flowers*, we are aware that the baseline performance is way behind what has been reported recently in [42, 59], in which low-level features are both RGB colors and dense SIFTs extracted at multiple scales. The purpose of using this dataset here is just to assess our method in the feature space on a fine-grained image classification dataset implemented within a simple but widely used baseline.

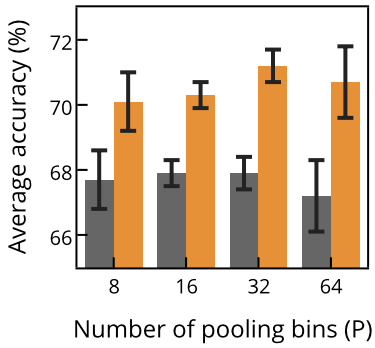
Comparison to a closely related work

We compare our method to [34], which is, to the best of our knowledge, the most notable work on

[2]Note that $P = 1$ means that no local pooling is conducted, i.e., global pooling (baseline). In other words, the image is represented with a traditional Bag of Features. This Bag of Features along with the image label are what passed to the classifier later.



(a) *Caltech-101* (b) *15 Scenes*



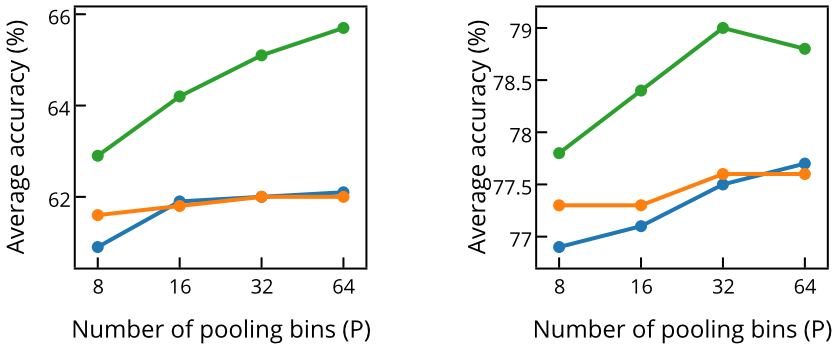
(c) *17 Flowers*

Figure 3.3: Classification accuracy (%) comparison between the method in previous work [34] (gray) and our method (orange). Our method outperforms [34] on all datasets for less feature dimensionality.

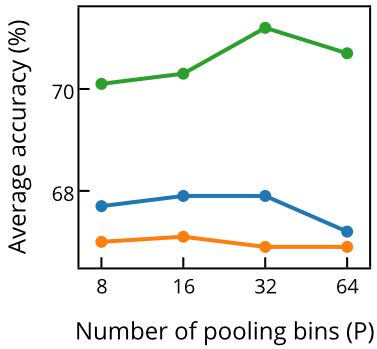
local pooling in the feature space within the adopted pipeline. This method relies on partitioning the feature space by clustering the input image's low-level descriptors over the codewords of a codebook obtained using k-means clustering and then jointly pooling only descriptors that belong to the same cluster, i.e., visually similar descriptors. Note that, in contrast to our method, this method partitions the feature space without any consideration of the semantics of the input image. Figure 3.3 compares the classification performances of the two methods on *Caltech-101*, *15 Scenes*, and *17 Flowers*.

The obtained results clearly show that our method outperformed [34] for all datasets. In fact, using only 8 bins, our method achieved better results even when 32 or 64 bins (whichever performed better) were utilized by the comparative method. The obtained results emphasize that our features are pooled from a space partitioning of a better quality than that of the comparative method.

It would be interesting to empirically assess the quality of the space partitioning utilized in the



(a) *Caltech-101* (b) *15 Scenes*



(c) *17 Flowers*

Figure 3.4: Classification accuracy (%) comparison among the method in previous work (blue), our method (green) and random (orange). On (a) and (b), features pooled from an appearance-based bins (previous work) almost always perform worse than those pooled from random bins.

two methods. To this end, we compared classification performance of features pooled from bins (space partitioning) obtained by three different methods: (1) Bregman co-clustering, (2) k-means, and (3) randomly selected from a k-means codebook of size 4096. The experiment was conducted on *Caltech-101*, *15 Scenes* and *17 Flowers* for $P \in \{8, 16, 32, 64\}$. The results obtained are shown in Figure 3.4. As expected, our features always outperformed randomly pooled ones. However, a more interesting finding is that on the first two datasets, features of [34] almost always performed worse (or similar to) than those pooled from random bins. This result is an evidence that k-means is far from providing an optimal partitioning of the feature space.

3.4.1.3 Results (2)

In this section, the proposed method is compared to other works on three datasets: *Caltech-101*, *15 Scenes*, and *Caltech-256*. We first compare Bregman pooling to other spatial pyramid (SP)-based methods. Then, the comparison is extended to state-of-the-art methods.

Comparison with SP-based methods

For a fair comparison, we implemented Bregman pooling within an SP following the previously explained details[3]. It should be noted that only on *caltech-256* we changed the adopted baseline and used the one described in [24]. The results obtained are shown in Table 3.1 for $P \in \{1, 8, 16\}$. Note that for $P = 1$, the proposed method reduces down to the SP model. We experimented with $P \in \{32, 64\}$ (not shown) and found that over-binning ($P > 16$) degrades the performance on all three datasets. This observation has been reported in [34]. Following the common practice of comparing obtained results to those of previous work [34, 56, 57, 60], Table 3.1 also quotes results reported for other SP-based methods.

However, since all quoted works are extensions to the original SP model of [22], simply listing the obtained results does not give a clear insight into how each improves the model. Thus, in order to avoid comparing apples to oranges, we break the listed works into four main groups based on what component, of the SP model, each improves. Thus, works are grouped into (1) those that improve the coding step, including works by [23], [24] and [60], (2) those that improve the pooling operator, including works by [23] and [42], (3) those that enrich the spatial information captured by the model, the works by [57, 61], and finally (4) those that locally pool in the feature space, including works by [34, 56], and ours. Table 3.1 also includes studies by [62] and [41], which are two widely cited benchmarking studies that extensively evaluated the model using different combinations of components and parameters. In the following, we discuss our obtained results within the context of each group individually.

Within the first group, [23] and [24] are highly successful extensions to the the SP model that adopt (aside from max pooling) two improved coding methods: SC and LLC coding, respectively. Our method was implemented within the former on the first two datasets and within the latter on the third

[3]The image's low-level features within each spatial cell (A total of 21 cells over 3 layers) are (1) clustered around the P pooling bins. Then, (2) pooled accordingly. The final image representation is the concatenation of all pooled features.

dataset. Thus, for a fair comparison with these extensions, we compared our best performance to our implementation of them (i.e., $P = 1$). We achieved 2.0% and 1.8% performance boosts over [23] on the first two datasets, and 0.7% performance boost over [24] on the third dataset. These results indicate the importance of our local pooling over these two SP extensions. Our method also outperformed the recent Collaborative Linear Coding (CLC) [60] on *15 Scenes* by 0.2% (but with +0.1 in standard deviation). However, due to the differences in experimental settings used (We used single-scale SIFTs and a 4096-dim codebook, while [60] used multi-scale SIFTs and a 2048-dim codebook.), it is difficult to compare the two precisely.

Within the second group, the AxMin@n pooling operator of [42] outperformed all other methods on *Caltech-101*. In fact, our best performance fell 2.5% behind their reported performance. However, it should be noted that [42] used dense SIFTs extracted at four different scales and thus each image is represented with an average number of 5200 descriptors. In any case, the results indicate the important role an adaptive pooling operator plays over the classification performance on this dataset. It is worth mentioning that within the same group, we obtained the best results on both *15 Scenes* and *Caltech-256*.

Our method also outperformed [57, 61] on all three datasets. However, it is worth mentioning that even with a relatively small feature dimension (smaller codebook) and less dense low-level features, [57] achieved a highly competitive result on *15 Scenes*.

The proposed method also outperformed [34, 56] on all datasets. Our better performance over [34] can be understood in light of the obtained results. However, a comparison with [56] is difficult due to the lack of (1) a public implementation of their method and (2) reported results over different datasets. However, we achieved 0.4% ($P = 8$) and 0.5% ($P = 16$) boosts in performance over their reported results on *Caltech-256*. Analyzing the significance of this boost is impossible as [56] did not report their standard deviation. One major drawback common to all methods within this group is the inflated feature dimension. This is inevitable as the feature space is partitioned within every cell of the pyramid. Although we report better performance than previous works for smaller feature dimensions, still our features have much larger dimensions than those of other SP-based methods.

Finally, it is worth mentioning that both AxMin@n pooling [42] and CLC coding [60] can be easily implemented within our method. Moreover, it would be interesting, in the future, to test how

Table 3.1: Average classification accuracy (%) comparison on *Caltech-101*, *15 Scenes*, and *Caltech-256* datasets.

Method	Caltech-101	15 Scenes	Caltech-256	Feature dimension
[22]	64.6 ± 0.8	81.4 ± 0.5	-	4200, 8400
[23]	73.2 ± 0.5	80.2 ± 0.9	40.1 ± 0.9	21504
[24]	73.4	-	47.7	43008, 86016
[62][a]	71.8 ± 1.0	84.1 ± 0.5	-	21504
[34]	77.3 ± 0.6	83.3 ± 1.0	41.6 ± 0.6[b]	1397760, 365568, 344064
[41]	76.1 ± 0.6	-	-	84000
[61]	67.1	82.5	-	5000
[42]	**81.3 ± 0.6**	-	-	86016
[60]	-	84.3 ± 0.2	-	43008
[56]	-	-	47.9	1134592
[57]	68.4	83.7	39.3[b]	13200, 13200, [c]
Proposed				
$P = 1$	76.8 ± 0.8	82.7 ± 0.3	47.7 ± 0.4	86016
$P = 8$	78.4 ± 0.8	84.3 ± 0.3	48.3 ± 0.3	774144
$P = 16$	78.8 ± 0.8	**84.5 ± 0.3**	**48.4 ± 0.3**	1462272

Works are listed in a chronological order.
Bold values indicate the best performance.
Some works do not report standard deviation.
A '-' means that the result is not reported in the corresponding work.
The feature dimension column lists dimension(s) of the image representation(s) used on the three datasets respectively.
[a] Intersection kernels are used rather than linear SVM.
[b] 30 training images per class are used.
[c] Feature dimension on *Caltech-256* is larger than 13200 but not clearly reported.

adopting either (or both) of them affects the classification performance of the proposed method.

Comparison with state-of-the-art methods

To complete the picture, Table 3.2 shows the best classification results obtained on *Caltech-101*, *15 Scenes* and *Caltech-256* of which we are aware. From Table 3.2, we can see that the three best performing methods [63–65] are all based on convolutional neural networks [30, 66]. By comparing Tables 3.1 and 3.2, we can see a huge gap separating SP-based methods from those based on convolutional neural networks.

In fact, convolutional neural networks have shown outstanding classification results on the ma-

Table 3.2: State-of-the-art methods on *Caltech-101*, *15 Scenes* and *Caltech-256*.

Dataset	Method	Result
Caltech-101	[63]	93.4 ± 0.5
15 Scenes	[64]	90.2 ± 0.3
Caltech-256	[65]	77.6 ± 0.1

jority of datasets recently. However, training convolutional neural networks requires huge amounts of data, time and processing power. For instance, [64] trained their network with more than 2.4 M images, and training took 6 days using a single GPU. On the other hand, [63] and [65] used 1.2 M images of ImageNet [67] as training data, and training the two networks took two weeks and three weeks, respectively.

3.4.2 Experiment (2)

3.4.2.1 Experimental protocol

Our experimental protocol is explained here. An overview of the used image datasets is given first, followed by an explanation of the implementation details.

Image datasets

We used two datasets already introduced in the previous chapter: *New York* and *Chicago* datasets. Individual datasets are briefly introduced in the following:

- *New York*: It consists of 14000 satellite images spanning three different classes representing three different road safety levels: low, neutral, and high. Images come in a fixed spatial resolution of 256×256 pixels each. This dataset was mined from over 647 thousand road accident reports collected by the New York Police Department (NYPD) over the period between 2012 and 2016.

- *Chicago*: It consists of 12000 satellite images spanning three different classes representing three different urban safety levels (crime rates): low, neutral, and high. Images come in a fixed spatial resolution of 256×256 pixels each. This dataset was mined from over 1 million crime incident reports collected by the Chicago Police Department over the period between 2001 and 2016.

Implementation details

Favoring the reproducibility of our results, the implementation details of our experiments are explained in this section.

- *Satellite imagery*: We used Google Static Maps API[4] to crawl all satellite images used in this experiment.
- *Pre-processing*: Images were first converted to grayscale.
- *Feature extraction & description*: Using VLFeat toolbox [54], low-level features were densely sampled over a rectangular grid of 16×16 pixel patches with a sampling rate of 4 pixels. Unless otherwise noted, a 128-dim SIFT descriptor was then computed for each extracted patch.
- *Codebooks*: Standard k-means clustering was used to generate codebooks. The number of codewords was always set to 4096.
- *Coding, pooling (operator), and normalization*: Unless otherwise noted, the combination of sparse coding and max pooling was used in our experiments. The final image representation is always ℓ_2-normalized.
- *Co-clustering*: We applied Bregman co-clustering offline on the training data of each dataset for a number of row clusters $P = \{8, 16\}$.
- *Spatial information*: We used a three-layer spatial pyramid of 21 cells (1×1, 2×2, 4×4) whenever spatial information was included.
- *Classification*: We adopted the one-versus-all methodology by training one SVM classifier per class using the library reported in [55]. The cost parameter was determined by cross-validation within the training data of the target dataset.
- *Evaluation*: We reported the average classification accuracy cross validated over three random 95%/5% training/testing data splits.

[4] https://developers.google.com/maps/documentation/static-maps

	x18	x19	x20
$P = 4$	0.454	0.461	0.436
$P = 8$	0.463	**0.465**	0.441

Table 3.3: Average prediction accuracy obtained for training six models obtained considering three different zoom levels (x18, x19, x20) and two different values of pooling bins ($P = \{4, 8\}$).

3.4.2.2 Results

We present the results of empirically evaluating the performance of the proposed framework (in chapter 2) using models learned from features engineered as explained in Sections 3.2 and 3.3 on two prediction tasks: road safety prediction and crime rate prediction.

Road safety prediction from satellite imagery

The purpose of this experiment is to evaluate the performance of the proposed flat architecture in learning models able to predict road safety from raw satellite images.

We have trained computational models on images of the *New York* dataset. Table 3.3 shows the average prediction accuracy obtained using 6 models obtained considering $P = \{4, 8\}$ pooling bins, and using satellite images captured at the three zoom levels (x18, x19, x20). From Table 3.3 we can make the following observations:

1. Flat models perform poorly in predicting road safety from raw satellite imagery for all studied P values and satellite imagery zoom levels.

2. The best performing model is the one trained on satellite imagery captured at zoom level x19 and using image representations pooled from 8 pooling bins.

3. Models trained on satellite images captured at zoom level x20 perform the worst for all P values.

4. Increasing the number of pooling bins P does not have a significant impact on prediction accuracy.

Results obtained in this experiment clearly demonstrate that flat models learned using the proposed architecture are far from being able to effectively predict road safety directly from raw satellite imagery.

	x18	x19	x20
$P = 4$	0.468	0.422	0.419
$P = 8$	**0.471**	0.427	0.420

Table 3.4: Average prediction accuracy obtained for training six models obtained considering three different zoom levels (x18, x19, x20) and two different values of pooling bins ($P = \{4, 8\}$).

Crime rate prediction from satellite imagery

The purpose of this experiment is to evaluate the performance of the proposed flat architecture in learning models able to predict crime from raw satellite images.

We have trained computational models on images of the *Chicago* dataset. Table 3.4 shows the average prediction accuracy obtained using 6 models obtained considering $P = \{4, 8\}$ pooling bins, and using satellite images captured at the three zoom levels (x18, x19, x20). From Table 3.4 we can make the following observations:

1. Flat models perform poorly in predicting crime rate from raw satellite imagery for all studied P values and satellite imagery zoom levels.

2. The best performing model is the one trained on satellite imagery captured at zoom level x18 and using image representations pooled from 8 pooling bins.

3. Models trained on satellite images captured at zoom level x20 perform the worst for all P values.

4. Increasing the number of pooling bins P does not have a significant impact on prediction accuracy.

Results obtained in this experiment clearly demonstrate that flat models learned using the proposed architecture are far from effective at predicting crime rate directly from raw satellite imagery.

3.5 Summary

In this chapter, we have proposed a novel feature-space local pooling method for the commonly adopted flat architecture of image classification. In contrast to methods in previous works, our method produces pooling bins that are aware of the semantic context of the input image within the dataset.

This is achieved by partitioning the feature space over clusters of visual prototypes common to images belonging to the same category (i.e., images of similar semantics). The clusters are obtained by Bregman co-clustering applied offline on a random subset of training data.

The proposed method was experimentally validated on four different datasets belonging to three different classification tasks. The results obtained demonstrate that (1) our method outperforms methods in previous works on local pooling in the feature space for less feature dimensionality and (2) when implemented within a spatial pyramid (SP), our method achieves comparable results on three of the datasets used.

Finally, we have empirically evaluated the performance of the proposed framework (of Chapter 2) using models learned using image representations engineered according to our proposed method. We have conducted two experiments covering two different public safety prediction tasks. The obtained results demonstrated that flat models perform poorly at predicting public safety from raw satellite imagery.

Chapter 4

Prediction Using Deep Models

4.1 Introduction

In this chapter we continue the empirical study we started in the previous chapter. We evaluate the performance of the proposed framework using models learned with deep Convolutional Neural Networks. The remainder of this chapter is organized as follows. Section 4.2 briefly introduces Convolutional Neural Networks. Section 4.3 presents and discusses the obtained experimental results. Finally, summary is given in Section 4.4.

4.2 Deep Image Classification Architecture

In this section, we briefly introduce Convolutional Neural Networks (ConvNets) and explain how we use them to train our deep models. It should be noted that this section is by no means intended to cover or fully introduce ConvNets and how they work. For more details on the subject, the reader is referred to [3, 68].

4.2.1 Convolutional Neural Networks

A ConvNet is a biology-inspired feedforward neural network that is designed to process data that come in multiple arrays, such as RGB color images. Similar to other deep learning approaches, ConvNets automatically learn from data hierarchical representations that capture patterns and statistics at multiple levels of abstraction.

Having their roots in the early neocognitron [69], ConvNets have been used in several applications since the early 1990s such as in [68]. Later in the 2000s, ConvNets proved highly successful in

several vision tasks where training examples are abundant. However, not until 2012 when trained on over a million images, ConvNets achieved a ground-breaking performance in generic object recognition. This success has since revolutionized the field of computer vision and pattern recognition, with ConvNets dominating most of the vision tasks nowadays [3].

A ConvNet takes a raw RGB image as an input and produces a class prediction as an output. Natural images are compositional hierarchies, in which lower level features combine to form higher level ones. ConvNets were designed to exploit this property. A typical ConvNet consists of a stack of convolutional layers followed by fully-connected layers ordered such that the output of one layer is the input of the next. A typical convolutional layer convolves a three-dimensional input tensor with a tensor of weights (filter maps). The weighted sum of the convolution is then passed through a nonlinearity function such as a Rectified Linear Unit (ReLU). The result is then passed through pooling operators to reduce the dimensionality of the representation and make it invariant to small perturbations. On the other hand, a fully-connected layer reduces the multidimensional input into a one-dimensional vector that is fed to a final classifier.

A ConvNet is trained end-to-end in a supervised fashion using Stochastic Gradient Descent (SGD) and backpropagation.

4.2.2 Model Learning

To train our models, we adopted transfer learning in which pre-learned knowledge is transferred from a source to a target problem. In our case, source and target problems are generic object/scene recognition, and road safety/crime rate prediction respectively. And the transferred knowledge is a set of low-level visual features such as edges and corners. In the deep learning community, this way of training is known as finetuning and it has been proven highly successful in augmenting learning when training data is limited [70, 71].

To finetune a pre-trained model, we first replaced the classification layer with a three-class output layer (representing the three safety labels). Weights of the added layer are randomly initialized, and the entire network is trained jointly using small learning rates.

4.3 Experimental Results

In this section, we present the results of two experiments we have conducted. In the first experiment, we evaluate the performance of deep models learned to predict public safety (road safety and crime rate) from raw satellite imagery. In the second experiment, we use the learned models from the previous experiment to generate city scale public safety maps predicted also from raw satellite imagery.

4.3.1 Experiment (1)

4.3.1.1 Experimental protocol

Our experimental protocol is explained here. An overview of the used image datasets is given first, followed by an explanation of the implementation details.

Image datasets

We used two datasets already introduced in the previous chapter: *New York* and *Chicago* datasets. Individual datasets are briefly introduced in the following:

- *New York*: It consists of 14000 satellite images spanning three different classes representing three different road safety levels: low, neutral, and high. Images come in a fixed spatial resolution of 256×256 pixels each. This dataset was mined from over 647 thousand road accident reports collected by the New York Police Department (NYPD) over the period between 2012 and 2016.

- *Chicago*: It consists of 12000 satellite images spanning three different classes representing three different urban safety levels (crime rates): low, neutral, and high. Images come in a fixed spatial resolution of 256×256 pixels each. This dataset was mined from over 1 million crime incident reports collected by the Chicago Police Department over the period between 2001 and 2016.

Implementation details

Favoring the reproducibility of the results, below we explain how experiments were implemented:

Satellite imagery: We used Google Static Maps API[1] to crawl all satellite images used in this experiment. Individual images have a spatial resolution of 256×256 pixels each.

[1] https://developers.google.com/maps/documentation/static-maps

	x18	x19	x20
ImageNet	0.740	0.766	0.739
Places205	0.755	0.775	0.745
ImageNet + Places205	0.778	**0.782**	0.771

Table 4.1: Average prediction accuracy obtained using nine models pre-trained on three different large-scale datasets and finetuned on satellite images captured at three different zoom levels.

ConvNet architecture: All ConvNets used in this experiments follow the AlexNet architecture [30] which is both simple and considered a landmark architecture.

Training: Our models were initialized from generic large-scale image datasets. Three datasets were considered: (1) ImageNet [72], (2) Places205 [64], and (3) both ImageNet and Places205 combined. Training was done using Caffe framework [73] run on a single Nvidia GeForce TITAN X GPU.

Evaluation: To evaluate the learned models, we reported the average prediction accuracy cross-validated on three random 5%/95% data splits. Reported results are obtained after 60,000 training iterations.

4.3.1.2 Results

We present the results of predicting road safety and crime rate from raw satellite imagery using deep models.

Road safety prediction from satellite imagery

The purpose of this experiment is twofold: (1) to investigate whether or not our assumption that visual features captured in satellite imagery can be effectively used as a proxy indicator of road safety. And (2) to evaluate the performance of ConvNets in learning deep models able to predict road safety from raw satellite images.

We have finetuned our ConvNet on images of the *New York* dataset. Table 4.1 shows the average prediction accuracy of nine models obtained considering three pre-training scenarios, and using satellite images captured at three zoom levels.

Spanning a range between 73.9% and 78.2%, the best performing model is the one obtained through finetuning a pre-trained model on both ImageNet and Places205 datasets using satellite images captured at zoom level x19. From Table 4.1, we make the following observations:

1. For all zoom levels, models pre-trained on both ImageNet and Places205 achieve the best, followed by models pre-trained on Places205, and finally models pre-trained on ImageNet. This is expected since satellite images have bird's eye/aerial viewpoint which makes them closer in composition to scene images of Places 205 rather than the object-centric images of ImageNet.

2. For all pre-training scenarios, finetuning using satellite images captured at zoom level x19 results in the best performance.

Results obtained in this experiment confirm our assumption that visual features captured in satellite imagery can be effectively used as a proxy indicator of road safety. Moreover, ConvNets are able to learn robust models that can predict road safety from raw satellite images.

Crime rate prediction from satellite imagery

Similarly, the purpose of this experiment is twofold: (1) to investigate whether or not our assumption that visual features captured in satellite imagery can be effectively used as a proxy indicator of crime rate. And (2) to evaluate the performance of ConvNets in learning deep models able to predict crime from raw satellite images.

The result of finetuning on our *Chicago* dataset is shown in Table 4.2. The table shows average prediction accuracy of twelve models obtained considering three pre-training scenarios using satellite images captured at four zoom levels.

Spanning a range between 63.8% and 79.5%, the best performing model is the one obtained through finetuning a pre-trained model on Places205 dataset using satellite images captured at zoom level x17. From Table 4.2, we make the following observations:

1. For all zoom levels (except zoom level x20), models pre-trained on Places205 perform the best, followed by models pre-trained on both Places205 and ImageNet, and finally models pre-trained on ImageNet. This is expected since satellite images have bird's eye/aerial viewpoint which makes them closer in composition to scene images of Places 205 rather than the object-centric images of ImageNet.

2. For all pre-training scenarios, models finetuned using satellite images captured at zoom level x17 perform the best. On the other hand, models finetuned on zoom level x20 images perform the worst.

	x17	x18	x19	x20
ImageNet	0.763	0.727	0.702	0.643
Places205	**0.795**	0.748	0.728	0.638
ImageNet + Places205	0.782	0.733	0.725	0.673

Table 4.2: Average prediction accuracy obtained using different models pre-trained on three different large-scale datasets and finetuned on satellite images captured at four different zoom levels.

Results obtained in this experiment confirm our assumption that visual features captured in satellite imagery can be effectively used as a proxy indicator of crime rate. Moreover, ConvNets are able to learn robust models that can predict crime rate from raw satellite images.

4.3.2 Experiment (2)

4.3.2.1 Experimental protocol

Our experimental protocol is explained here. An overview of the used image datasets is given first, followed by an explanation of the implementation details.

Image datasets

We used three datasets already introduced in Chapter 2: *Denver 1*, *Denver 2* and *San Francisco*. Individual datasets are briefly introduced in the following:

- *Denver 1*: It consists of 21406 satellite images spanning three different classes representing three different road safety levels: low, neutral, and high. This dataset was mined from over 110 thousand road accident reports collected by the Denver Police Department over the period between 2013 and 2016.

- *Denver 2*: It consists of 25169 satellite images spanning three different classes representing three different urban safety levels (crime rates): low, neutral, and high. This dataset was mined from over 198 thousand crime incident reports collected by the denver Police Department over the period between 2014 and 2016.

- *San Francisco*: It consists of 19897 satellite images spanning three different classes representing three different urban safety levels (crime rates): low, neutral, and high. This dataset was

mined from over 652 thousand crime incident reports collected by the San Francisco Police Department (SFPD) over the period between 2003 and 2016.

Implementation details

Favoring the reproducibility of the results, below we explain how experiments were implemented:

Satellite imagery: We used Google Static Maps API[2] to crawl all satellite images used in this experiment. Individual images have a spatial resolution of 256×256 pixels each.

Prediction model: We used the best performing model for each task from the previous experiment.

Evaluation: We evaluated the quality of the predicted maps by reporting the average prediction accuracy calculated across all classes.

4.3.2.2 Results

We present the results of investigating the reusability of the learned deep models (of the previous experiment) across different cities.

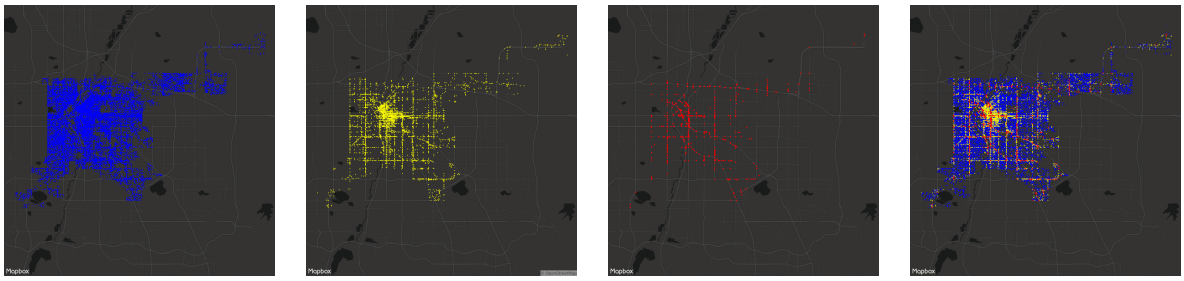
Road safety mapping

The purpose of this experiment is to empirically evaluate the reusability of the learned deep models. To this end, we used New York models to generate a city-scale road safety map predicted from raw satellite imagery for the city of Denver

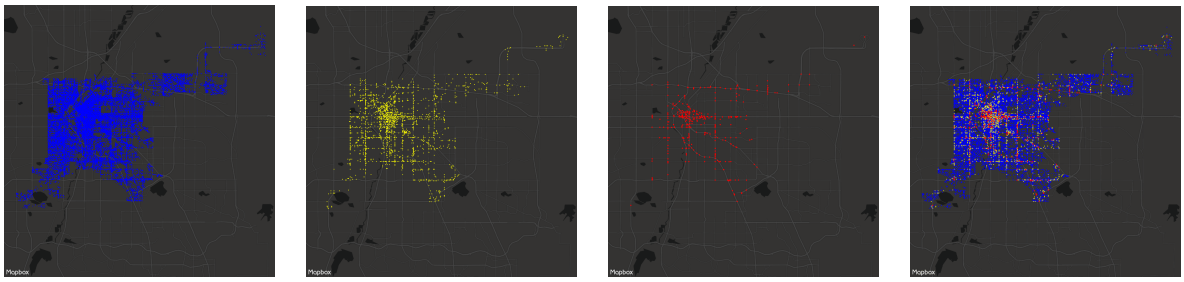
To this end, we used the best performing model learned from New York city to predict safety labels of the 21,406 images of *Denver 1* dataset. Figure 4.1 shows a city-scale road safety map for the city of Denver. The upper row is a map made from 110,870 traffic-accident reports collected by the Denver police department over the period between July 2013 and July 2016. The bottom row shows a map predicted completely from raw satellite images. The first three columns (left to right) illustrate the three safety levels (high: blue, neutral: yellow, and low: red) mapped individually. The fourth column illustrates all safety levels mapped together. Compared to the official map (upper row), the predicted map (bottom row) has an accuracy of 73.1%.

Denver city and New York city are quite different from each other in terms of the level of development, area, population, traffic, etc. Thus, demonstrating that a model learned from New York city data can effectively predict road safety in Denver city proves that models are practically reusable (to a

[2]https://developers.google.com/maps/documentation/static-maps



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



(d) (e) (f) (g) 73.1%

Figure 4.1: City-scale map of Denver city indicating road safety in three different levels (high: blue, neutral: yellow, and low: red). Upper row is a map made from data collected by Denver city Police Department between July 2013 and July 2016. Bottom row is a map predicted from raw satellite imagery using our approach. First three columns (left to right) represent the three safety levels mapped individually. The fourth column represents all safety levels mapped together. This figure is best viewed in digital format.

certain degree). Moreover, in order to quantify the accuracy of the predicted map, we had to choose a city that has its official traffic-accident reports publicly accessible so that we can compare our results to a ground truth. Therefore, for the previous reasons we chose Denver city to map in this experiment.

Results obtained in this experiment confirm that deep models learned from road safety data collected in a large city can be reused to predict road safety in smaller cities with less resources.

Crime mapping

The purpose of this experiment is to empirically evaluate the reusability of the learned deep models. To this end, we applied Chicago models to generate city-scale crime maps predicted from raw satellite imagery for two US cities, namely Denver and San Francisco.

We used the best performing Chicago models to predict labels of the 25169 images of the *Denver 2* dataset. Figure 4.2(a-g) shows a city-scale crime map for the city of Denver. The upper row is a map made from 198506 crime reports collected by the Denver police department over the period between

July 2014 and July 2016. The bottom row shows a map predicted completely from raw satellite images. Compared to the official map (upper row), the predicted map (bottom row) has an accuracy of 72.7%.

We also predicted the labels of the 19897 images of the *San Francisco* dataset. Figure 4.2(h-o) shows a city-scale crime map for the city of San Francisco. The upper row is a map made from 652,807 crime reports collected by the San Francisco police department over the period between March 2003 and September 2016. The bottom row shows a map predicted completely from raw satellite images. Compared to the official map (upper row), the predicted map (bottom row) has an accuracy of 70.8%.

For both maps, the first three columns (left to right) illustrate the three crime rate labels (low: blue, neutral: yellow, and high: red) mapped individually. The fourth column illustrates the three labels mapped together.

Since Chicago is quite different from both Denver and San Francisco in terms of population, area, and crime rate, demonstrating that a model learned from data collected in Chicago can effectively (to a certain degree) predict crime in both Denver and San Francisco proves that our learned models are practically reusable. Moreover, in order to quantify the accuracy of the predicted maps, we had to choose cities that have their official crime data publicly accessible so that we can compare our results to a ground truth. On the basis of these criteria we have decided to map the cities of Denver and San Francisco in this experiment.

Results obtained in this experiment confirm that deep models learned from crime data collected in one city can be reused in different cities.

4.4 Summary

In this chapter we have continued the empirical study we started in the previous chapter. We have evaluated the performance of the proposed framework (of Chapter 2) using models learned with deep Convolutional Neural Networks (ConvNets). The obtained results demonstrated that deep models perform reasonably well at predicting public safety from raw satellite imagery.

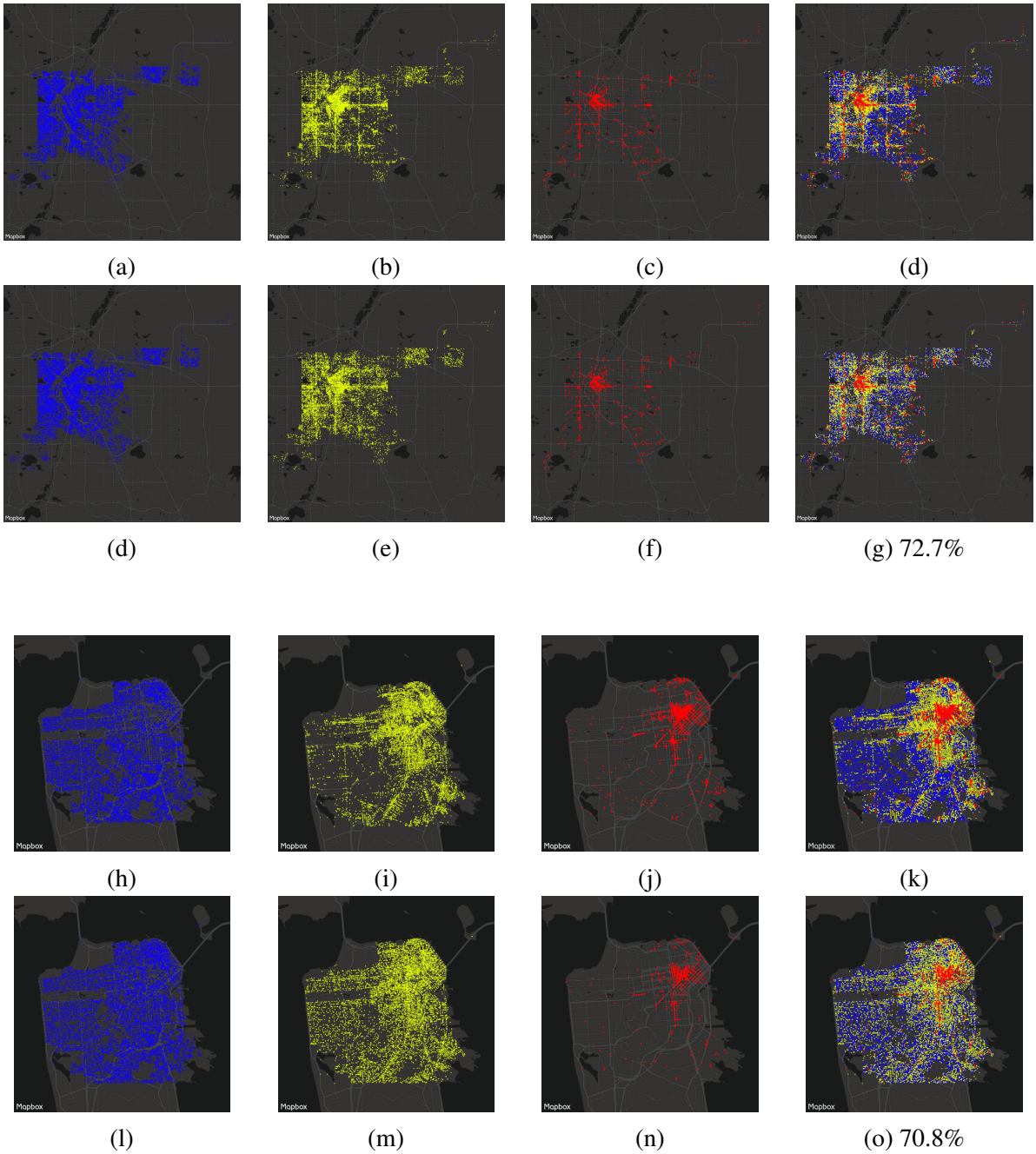


Figure 4.2: City-scale crime maps of the cities of Denver (a-g) and San Francisco (h-o). For each city, the upper row is a map made from official data. While, the bottom row is a map completely predicted from raw satellite imagery. First three columns (left to right) represent the three crime rate labels (low: blue, neutral: yellow, and high: red) mapped individually. The fourth column represents all labels mapped together. The predicted maps have an accuracy of 72.7% and 70.8%, respectively. Best viewed in digital format.

Chapter 5

Summary and Future Work

5.1 Summary

In this study, we have discussed our approach to public safety prediction directly from raw satellite imagery using tools from modern machine learning and computer vision. Our approach is applied at a city scale thus allowing for the automatic generation of city-scale public safety maps. In this work we focused our attention on two types of public safety maps, namely road safety maps and crime maps. We formalized the problem of public safety mapping as a supervised image classification problem, in which a city-scale satellite map is treated as a set of satellite images each of each is assigned a safety label predicted using a model learned from training samples. To obtain this training data we leveraged official police reports collected by police departments and released as open data. The idea is to mine large-scale datasets of official police reports for high-resolution satellite images labeled with safety scores calculated based on number and severity/category of incidents. We validated and tested the robustness of the learned models for both road safety and crime rate prediction tasks over four different US cities, namely New York, Chicago, San Francisco, and Denver. We also attempted to investigate the reusability of the learned computational models across different cities.

Main contributions made in this thesis can be summarized as follows: (1) proposing a framework for automatic city-scale public safety prediction from satellite imagery, (2) proposing an automatic approach for obtaining labeled satellite imagery via mining large-scale collections of official police reports released as open data, and (3) introducing five labeled satellite imagery datasets representing four different US cities, and mined from over 2.5 million official police reports.

As an empirical validation, we have conducted an extensive experimental study as detailed in

chapters 3 and 4. In this study we have trained computational models on satellite images mined from over 2.5 million official police reports collected by four different police departments in the US and released as open data. These models were learned using two different image classification architectures: (1) flat SVM-based architecture, and (2) deep ConvNet-based architecture. Obtained results can be summarized as follows. Deep models outperform flat models which perform poorly. Our best performing models are able to predict road safety and crime rate from raw satellite imagery with an accuracy that reaches up to 79%. Models learned from data collected in one city can be effectively (to a certain degree) reused across different cities. These results prove our assumption that visual information contained in satellite imagery has the potential to be used as an effective proxy indicator of public safety.

5.2 Future Work

Although this thesis introduces a proof-of-concept study on predicting public safety at a city-scale using affordable and accessible tools and data sources (targeting cities where proper data collection is not affordable), our study suffers from several limitations. First, our models do not take crime category or accident severity level into consideration. We have used crime incident/road accident count only as safety scores. We believe that training models on more elaborate data will result in more insightful maps. Second, our models predict public safety without taking time into consideration. In other words, our maps do not differentiate between day and night or summer and winter. Third, although we proved our method effective (to a certain degree) in predicting public safety in several US cities (target cities) using models trained on data collected in Chicago and New York (source cities), we have not considered a more extreme case in which both cities are located in two different continents (e.g., source city: Chicago. Target city: Nairobi) where architecture, city planning, level of development, etc. differ extremely. These limitations among others are to be addressed in future work.

Bibliography

[1] M. Leidig, R. M. Teeuw, and A. D. Gibson, "Data poverty: A global evaluation for 2009 to 2013-implications for sustainable development and disaster risk reduction," *International Journal of Applied Earth Observation and Geoinformation*, vol. 50, pp. 1–9, 2016.

[2] J. Dash and B. O. Ogutu, "Recent advances in space-borne optical remote sensing systems for monitoring global terrestrial ecosystems," *Progress in Physical Geography*, vol. 40, no. 2, pp. 322–351, 2016.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] S.-P. Miaou, J. J. Song, and B. K. Mallick, "Roadway traffic crash mapping: a space-time modeling approach," *Journal of Transportation and Statistics*, vol. 6, pp. 33–58, 2003.

[5] S. Chainey and J. Ratcliffe, *GIS and crime mapping*. John Wiley & Sons, 2013.

[6] D. Dietrich, J. Gray, T. McNamara, A. Poikola, P. Pollock, J. Tait, and T. Zijlstra, "Open data handbook," 2009.

[7] M. G. Maxfield, "The national incident-based reporting system: Research and policy applications," *Journal of Quantitative Criminology*, vol. 15, no. 2, pp. 119–149, 1999.

[8] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA., 1967, pp. 281–297.

[9] G. F. Jenks, "The data model concept in statistical mapping," *International yearbook of cartography*, vol. 7, no. 1, pp. 186–190, 1967.

[10] Q. Chen, X. Song, H. Yamada, and R. Shibasaki, "Learning deep representation from big and heterogeneous data for traffic accident inference," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[11] T. K. Anderson, "Kernel density estimation and k-means clustering to profile road accident hotspots," *Accident Analysis & Prevention*, vol. 41, no. 3, pp. 359–364, 2009.

[12] M. Bíl, R. Andrášik, and Z. Janoška, "Identification of hazardous road locations of traffic accidents by means of kernel density estimation and cluster significance evaluation," *Accident Analysis & Prevention*, vol. 55, pp. 265–273, 2013.

[13] Z. Xie and J. Yan, "Detecting traffic accident clusters with network kernel density estimation and local spatial statistics: an integrated approach," *Journal of transport geography*, vol. 31, pp. 64–71, 2013.

[14] Q. Han, Y. Zhu, L. Zeng, L. Ye, X. He, X. Liu, H. Wu, and Q. Zhu, "A road hotspots identification method based on natural nearest neighbor clustering," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 553–557.

[15] N. Naik, J. Philipoom, R. Raskar, and C. Hidalgo, "Streetscore–predicting the perceived safety of one million streetscapes," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2014, pp. 793–799.

[16] R. Herbrich, T. Minka, and T. Graepel, "Trueskill: A bayesian skill rating system," in *Advances in neural information processing systems*, 2006, pp. 569–576.

[17] A. Dubey, N. Naik, D. Parikh, R. Raskar, and C. A. Hidalgo, "Deep learning the city: Quantifying urban perception at a global scale," in *European Conference on Computer Vision*. Springer, 2016, pp. 196–212.

[18] K. Kianmehr and R. Alhajj, "Effectiveness of support vector machine for crime hot-spots prediction," *Applied Artificial Intelligence*, vol. 22, no. 5, pp. 433–458, 2008.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[20] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005, pp. 886–893.

[21] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.

[22] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 2169–2178.

[23] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009, pp. 1794–1801.

[24] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3360–3367.

[25] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision (IJCV)*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[26] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *Journal of Physiology*, vol. 160, pp. 106–154, 1962.

[27] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.

[28] J. Sivic and A. Zisserman, "Video Google: a text retrieval approach to object matching in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003, pp. 1470–1477.

[29] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision (ECCV)*, 2004, pp. 1–22.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[31] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, no. 8, pp. 1798–1828, 2013.

[32] Y. Jia, C. Huang, and T. Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3370–3377.

[33] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei, "Object-centric spatial pooling for image classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012, pp. 1–15.

[34] Y.-L. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: multi-way local pooling for image recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011, pp. 2651–2658.

[35] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 9, pp. 1704–1716, Sept 2012.

[36] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, "Image classification using super-vector coding of local image descriptors," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010, pp. 141–154.

[37] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha, "A generalized maximum entropy approach to Bregman co-clustering and matrix approximation," *Journal of Machine Learning Research (JMLR)*, vol. 8, pp. 1919–1986, 2007.

[38] Y.-L. Boureau, "Learning hierarchical feature extractors for image recognition," Ph.D. dissertation, New York University, 2012.

[39] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *International Journal of Computer Vision (IJCV)*, vol. 65, no. 1-2, pp. 43–72, Nov. 2005.

[40] S. Avila, N. Thome, M. Cord, E. Valle, and A. De A. AraúJo, "Pooling in image representation: The visual codeword point of view," *Computer Vision and Image Understanding (CVIU)*, vol. 117, no. 5, pp. 453–465, 2013.

[41] 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 (BMVC)*, 2011, pp. 76.1–76.12.

[42] P. Koniusz, F. Yan, and K. Mikolajczyk, "Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection," *Computer Vision and Image Understanding (CVIU)*, vol. 117, no. 5, pp. 479–492, 2013.

[43] V. N. Vapnik, *Statistical learning theory*, 1st ed. John Wiley and Sons, Inc, 1998.

[44] J. A. Hartigan, "Direct clustering of a data matrix," *Journal of the American Statistical Association*, vol. 67, no. 337, pp. 123–129, 1972.

[45] J. Liu and M. Shah, "Scene modeling using co-clustering," in *Proceedings of the International Conference on Computer Vision (ICCV)*, Oct 2007, pp. 1–7.

[46] A. Gupta and R. Bowden, "Unity in diversity: Discovering topics from words: Information theoretic co-clustering for visual categorization," in *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, 2012, pp. 628–633.

[47] L. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 3, pp. 200 – 217, 1967.

[48] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2003, pp. 89–98.

[49] 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," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2004, pp. 178–178.

[50] G. Griffin, A. Holub, and P. Perona, "The Caltech 256," California institute of technology, Tech. Rep., 2007.

[51] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision (IJCV)*, vol. 42, no. 3, pp. 145–175, 2001.

[52] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005, pp. 524–531.

[53] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 1447–1454.

[54] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[55] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research (JMLR)*, vol. 9, pp. 1871–1874, Jun. 2008.

[56] S. Fanello, N. Noceti, C. Ciliberto, G. Metta, and F. Odone, "Ask the image: Supervised pooling to preserve feature locality," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 851–858.

[57] R. Khan, C. Barat, D. Muselet, and C. Ducottet, "Spatial histograms of soft pairwise similar patches to improve the bag-of-visual-words model," *Computer Vision and Image Understanding (CVIU)*, vol. 132, no. 0, pp. 102–112, 2015.

[58] C. Wang and K. Huang, "How to use bag-of-words model better for image classification," *Image and Vision Computing*, 2014.

[59] Q. Chen, Z. Song, Y. Hua, Z. Huang, and S. Yan, "Hierarchical matching with side information for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 3426–3433.

[60] Z. Wang, J. Feng, and S. Yan, "Collaborative linear coding for robust image classification," *International Journal of Computer Vision (IJCV)*, pp. 1–12, 2014.

[61] R. Khan, C. Barat, D. Muselet, C. Ducottet, F. Saint-Etienne, and F. Etienne, "Spatial orientations of visual word pairs to improve bag-of-visual-words model," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2012, pp. 102–112.

[62] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 2559–2566.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014, pp. 346–361.

[64] 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 (NIPS)*, 2014, pp. 487–495.

[65] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," 2014.

[66] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[67] 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," 2014.

[68] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[69] K. Fukushima and S. Miyake, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern recognition*, vol. 15, no. 6, pp. 455–469, 1982.

[70] S. Branson, G. Van Horn, S. Belongie, and P. Perona, "Bird species categorization using pose normalized deep convolutional nets," *arXiv preprint arXiv:1406.2952*, 2014.

[71] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller, "Recognizing image style," *arXiv preprint arXiv:1311.3715*, 2013.

[72] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[73] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

Publications by the Author

- **Journal Papers**

  1. A. Najjar, T. Ogawa, and M. Haseyama. Bregman pooling: feature-space local pooling for image classification. *International Journal of Multimedia Information Retrieval (IJMIR),* vol. 4, no. 4, pp. 247-259. December 2015.

- **International Conference Papers**

  1. A. Najjar, S. Kaneko, and Y. Miyanaga. Crime mapping from satellite imagery via deep learning. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV).* March 2017.
  2. A. Najjar, S. Kaneko, and Y. Miyanaga. Combining satellite imagery and open data to map road safety. In *Proceedings of the 31st Conference on Artificial Intelligence (AAAI),* February 2017. (Acceptance rate: 24.6%)
  3. A. Najjar, T. Ogawa, and M. Haseyama. Recoverable projection based dimensionality reduction and the use of fractional distance measures in large scale mobile visual search. In *Proceedings of the 28th International Technical Conference on Circuits, Systems, Computers and Communications (ITC-CSCC)*, pp. 842-845. July 2013.
  4. A. Najjar, T. Ogawa, and M. Haseyama. Dimensionality reduction of sparse visual features via recoverable projection for large scale mobile visual search. In *Proceedings of the International Workshop on Advanced Image Technology (IWAIT)*, pp. 278-282. January 2013.

- **Domestic Conference Papers**

1. A. Najjar, S. Kaneko, and Y. Miyanaga. Road safety prediction from satellite imagery via deep learning. In *Proceedings of the 27th Vision Engineering Workshop (ViEW),* December 2016.
2. A. Najjar, T. Ogawa, and M. Haseyama. A note on compacting sparse visual features via recoverable projection for large scale mobile visual search. Shibukai. October 2012.