

Washington University in St. Louis
Washington University Open Scholarship

Engineering and Applied Science Theses &
Dissertations

Engineering and Applied Science


Winter 12-2016

Indoor Scene Localization to Fight Sex Trafficking in Hotels

Abigail Stylianou

Washington University in St. Louis

Follow this and additional works at: http://openscholarship.wustl.edu/eng_etds

 Part of the [Artificial Intelligence and Robotics Commons](#), [Engineering Commons](#), and the [Forensic Science and Technology Commons](#)

Recommended Citation

Stylianou, Abigail, "Indoor Scene Localization to Fight Sex Trafficking in Hotels" (2016). *Engineering and Applied Science Theses & Dissertations*. 198.

http://openscholarship.wustl.edu/eng_etds/198

This Thesis is brought to you for free and open access by the Engineering and Applied Science at Washington University Open Scholarship. It has been accepted for inclusion in Engineering and Applied Science Theses & Dissertations by an authorized administrator of Washington University Open Scholarship. For more information, please contact digital@wumail.wustl.edu.

Washington University in St. Louis
School of Engineering and Applied Science
Department of Computer Science and Engineering

Thesis Examination Committee:
Robert Pless, Chair
Yasutaka Furukawa
Sanmay Das

Indoor Scene Localization to Fight Sex Trafficking in Hotels

by

Abigail Stylianou

A thesis presented to the
Graduate School of Arts and Sciences
of Washington University in
partial fulfillment of the
requirements for the degree
of Master of Science

December 2016
Saint Louis, Missouri

copyright by
Abigail Stylianou
2016

Contents

List of Figures	iii
Acknowledgments	iv
Abstract	v
1 Introduction	1
2 Dataset Creation Application Details	4
2.1 Publicly Available Imagery	4
2.2 Crowd-sourced Image Collection	5
2.3 Dataset Scope	6
2.4 Implementation Details	6
2.5 Application Statistics	7
3 Feature Matching Approaches	10
3.1 k-Nearest Neighbor Matching with SIFT Features	10
3.2 k-Nearest Neighbor Matching with Learned Features	11
3.3 Bag of Words Matching with SIFT Features	12
3.4 Vector of Locally Aggregated Descriptors (VLAD) Matching	13
4 Experimental Results	14
4.1 Matching Between Expedia Images	14
4.2 Matching TraffickCam to Expedia Images	15
5 Conclusions and Future Work	20
References	23

List of Figures

1.1	System to collect hotel room images and match to law enforcement queries. .	3
2.1	Smartphone app screenshots.	5
2.2	Example images from TraffickCam and Expedia.	8
2.3	Number of images per hotel.	9
2.4	Number of TraffickCam users per hotel.	9
4.1	Matching performance by feature type.	17
4.2	Examples of successes and failures in scene recognition using SIFT features. .	18
4.3	Examples of successes and failures in scene recognition using NetVLAD features.	19

Acknowledgments

I would like to thank my husband and parents for their unending support, my beautiful daughter for making the whole world brighter, and my advisor, Dr. Robert Pless, for allowing me to explore solutions to real and meaningful problems in the world. I would also like to acknowledge our partners on this project including Dr. Richard Souvenir at Temple University, Jon Brandt at Adobe, the Exchange Initiative, and the Congregation of St. Joseph.

Abigail Stylianou

WASHINGTON UNIVERSITY IN SAINT LOUIS
December 2016

ABSTRACT OF THE THESIS

Indoor Scene Localization to Fight Sex Trafficking in Hotels

by

Abigail Stylianou

Master of Science in Computer Science

Washington University in St. Louis, December 2016

Research Advisor: Dr. Robert Pless

Images are key to fighting sex trafficking. They are: (a) used to advertise for sex services, (b) shared among criminal networks, and (c) connect a person in an image to the place where the image was taken. This work explores the ability to link images to indoor places in order to support the investigation and prosecution of sex trafficking. We propose and develop a framework that includes a database of open-source information available on the Internet, a crowd-sourcing approach to gathering additional images, and explore a variety of matching approaches based both on hand-tuned features such as SIFT and learned features using state of the art deep learning approaches. We concentrate on spatio-temporal indexing of hotel rooms, and to date have an index of more than 1.5 million geo-coded images. Our smart-phone app collects contextual information and metadata alongside images.

Chapter 1

Introduction

Images are key to fighting sex trafficking. They are: (a) used to advertise for sex services, (b) shared among criminal networks, and (c) connect a person in an image to the place where the image was taken. This work explores the ability to link images to indoor places in order to support the investigation and prosecution of sex trafficking. We propose and develop a framework that includes a database of open-source information available on the Internet, a crowd-sourcing approach to gathering additional images, and explore a variety of matching approaches based both on hand-tuned features such as SIFT and learned features using state of the art deep learning approaches. We concentrate on spatio-temporal indexing of hotel rooms, and to date have an index of more than 1.5 million geo-coded images. Our smart-phone app collects contextual information and metadata alongside images.

Images are a common way to advertise sex services. Images are interesting from an investigative standpoint because they connect the person in the image to the location where the image was taken. Therefore, they can help to characterize where a particular person was at different times. In the context of a sex trafficking investigation, this can be used to directly confirm that a person was in different states or countries. Among other things, this can change the set of laws under which a trafficker can be prosecuted.

We build a dataset from publicly shared imagery on hotel booking sites, as well as from a smartphone app to crowd-sourcing the collection of pictures of hotel rooms. The crowd-sourcing option takes advantage of large scale trends in how people use social media; approximately 350 million photos are uploaded daily to Facebook. Tapping into this already common behavior creates the potential to rapidly create a relatively comprehensive, distributed, and continually updated resource that details the current appearance of hotel rooms worldwide.

It also permits exploration into creating apps that encourage the acquisition of the most useful pictures for the matching process.

We know of no prior efforts to explicitly match images to hotel rooms. Informal discussion with investigators reveal that the most commonly used technologies are manual searches through possible hotels in an area of interest, or using tools like Google "search by image" which returns images that are visually similar to a query. While Google does not advertise its proprietary searching method, there is a rich literature on content based image retrieval, and recent research on methods that scale up to large image database sizes [16]. In our paper, we explore baseline approaches based on SIFT features [15], and convolutional neural networks [13].

Our work seeks approaches to create Internet tools to fight sex-trafficking. It is widely understood that cyber-space markets are challenging places to coordinate efforts, because there is an asymmetry of incentives in cyber-space markets. Technology can be easily exploited by sex-traffickers to coordinate activities and advertising services. In contrast, the anti-trafficking efforts are sometimes hampered because the incentives for non-profit organizations often make them less likely to freely share resources as they struggle for recognition and funding for their efforts [4].

Open data has been used in trying to estimate the prevalence of trafficking [7] and to determine the effectiveness of US anti-trafficking funding projects [8]. Technology efforts have focussed on create search tools that index open data in different forms to create an interface that can be used for query and analysis [12, 19, 17], but to our knowledge there has not be any system that is explicitly focused on imagery.

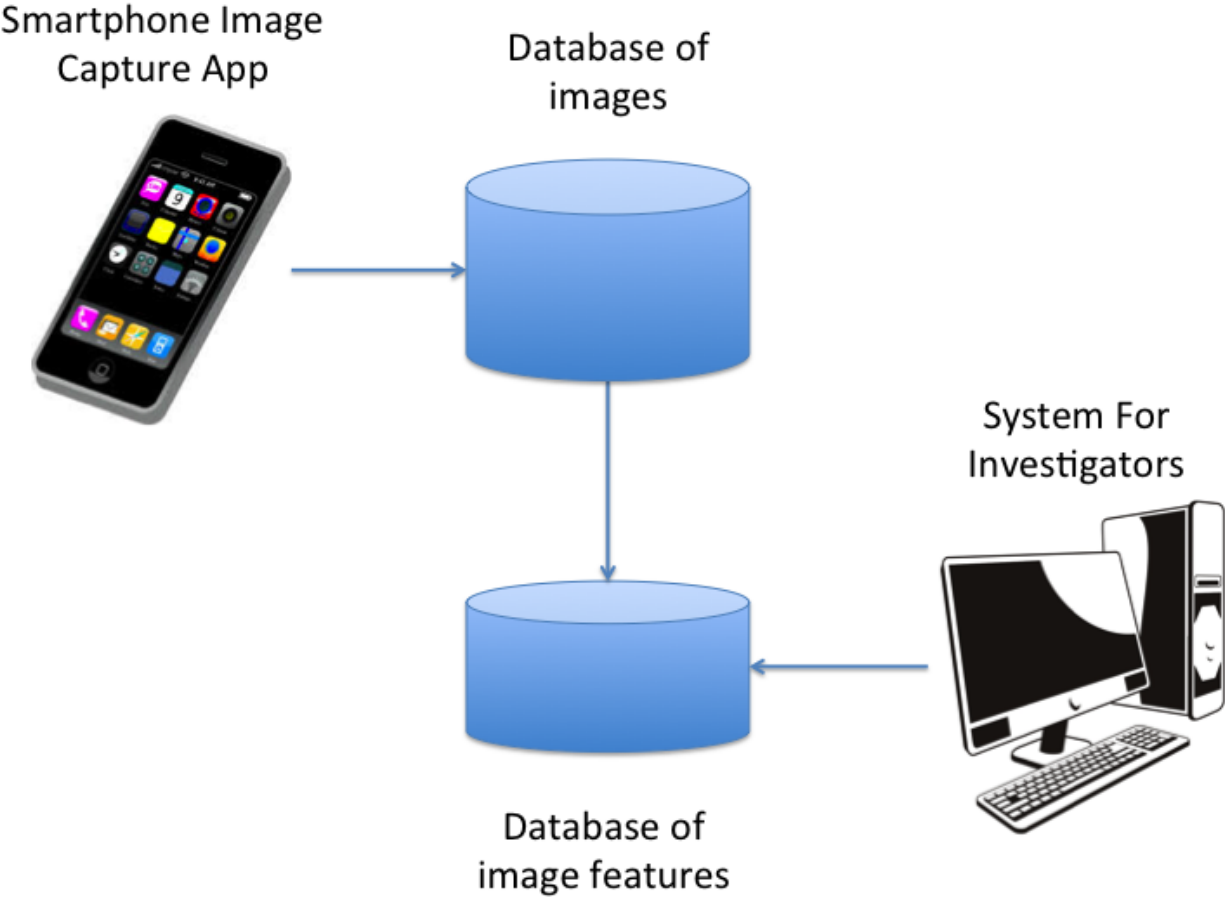


Figure 1.1: A database of carefully annotated pictures of hotel rooms offers an important investigative tool. We explore frameworks that combine a crowd-sourcing mobile phone app that contributes to a database of images with a system that computes features from each image and offers an interface to find match query images to similar images in the database.

Chapter 2

Dataset Creation Application Details

In order to have the highest likelihood of finding a good feature match between an investigator’s query image and the images in our dataset, our dataset should have as many images of as many rooms in as many hotels as possible. Additionally, it should have images from as many different times as possible. Hotels regularly renovate and change their internal appearance, meaning that photographs in our dataset may become outdated. These outdated images may still be valuable, however, in pinpointing the time frame in which an individual was trafficked (e.g., “This photograph was taken before the 2015 renovations, which means the person in the photograph was a minor at the time the advertisement was placed.”).

2.1 Publicly Available Imagery

We keep track of the millions of images made available through Expedia’s Affiliate Network API (<http://developer.ean.com/>) and create a reference in our database to the original data and its associated metadata. These photos, however, are often provided by the hotels themselves and may not present a comprehensive view of the hotel itself (e.g., only the nicest rooms from good angles in the best lighting). They may also not be updated following renovations. Both of those flaws would be problematic if these photographs were the only representations our dataset had of these hotels.

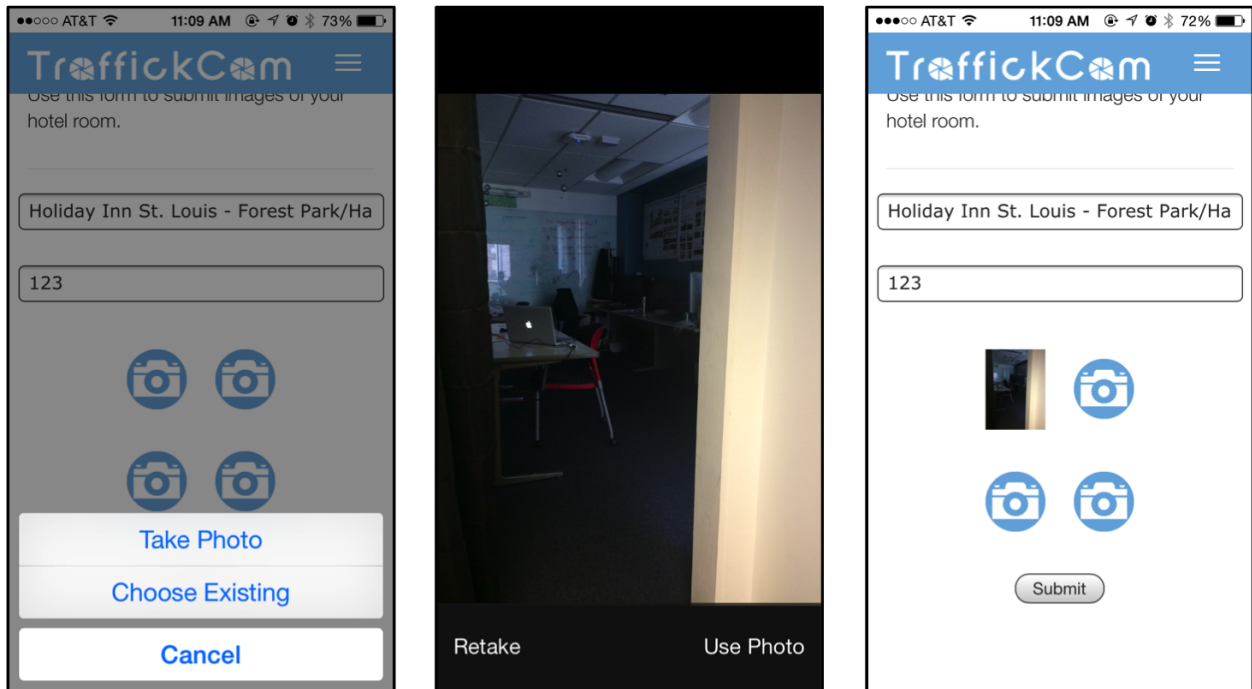


Figure 2.1: Screenshots of the smartphone app, TraffickCam, that allows anyone to contribute to the database. The app is designed to require minimal user time and to protect the user’s identity.

2.2 Crowd-sourced Image Collection

To supplement the images captured from existing datasets, we have created a smartphone crowd-sourcing application named TraffickCam, which allows travellers to upload their own photographs of a hotel room. This application is shown in Figure 2.1. Users are asked to provide minimal information regarding the photo – the name of the hotel they’re staying in and their room number, along with images of the room.

The application, called TraffickCam, is available from the iOS and Android stores, in addition to being accessible via any modern browser at <http://traffickcam.org>.

Examples of images from both the Expedia dataset and the TraffickCam dataset, as well as representative images that law enforcement might upload to the TraffickCam system can be seen in Figure 2.2.

2.3 Dataset Scope

The present TraffickCam database includes 1,629,505 images from 150,289 unique hotels. Of these hotels, 131,244 hotels have only Expedia images, 15,242 have only TraffickCam images, and 10,742 hotels have both TraffickCam and Expedia images. Figure 2.3 shows a histogram of the number of images per hotel. For TraffickCam, the most common number of images per hotel are in increments of four, due to the app requesting four images at a time (but allowing any number between one and four).

While the TraffickCam application purposefully collects no identifiable information about users to protect them from any legal action, we are able to estimate the number of users per hotel by the timestamp of the images uploaded – the application asks users for a set of four images, so we assume images that are disjoint in time are from different users. The plot in Figure 2.4 shows that most TraffickCam hotels have only a user or two, while a few have many more users. These hotels with many users are largely locations where TraffickCam training events have been held.

2.4 Implementation Details

We have implemented a RESTful API in Python Django, a web framework for rapid web development. Django handles the interaction between the server side code, web front end code, MySQL database and Apache web server. Test, stage and production Ubuntu environments are hosted through Amazon Web Services.

The iOS app, available through the Apple App Store, is simply a container that renders an HTML5+jQuery+AJAX web application hosted on <https://traffickcam.org>, rather than a full native application. This allows for rapid development and easy exploration of different user experience choices (e.g., different motivational messages to display to users upon submission). The Android application is a native application available on the Google Play store.

2.5 Application Statistics

Since TraffickCam was released in December of 2015, there have been 68,700 installations on iOS devices and 28,500 installations on Android devices. These installations are primarily from users in the United States, where the search tool will first be deployed for law enforcement, but also include several thousand installations each from Europe and Asia. On average since the advertised release of the TraffickCam applications for iOS and Android in June of 2016, users have submitted just over 530 images a day.

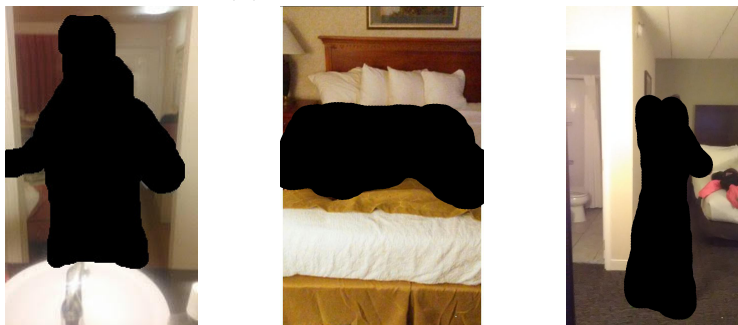
The search interface based on the methods detailed in this work is currently being evaluated by St. Louis County Police Department.



(a) Expedia Images



(b) TraffickCam Images



(c) Example Censored Query Images from Law Enforcement

Figure 2.2: The top set of images are from Expedia and the middle set of images taken by TraffickCam users at the same hotel. The bottom set of images are censored versions of the types of images that might be provided by law enforcement. These examples demonstrate the discrepancy in the types of photos provided by Expedia, by the TraffickCam app and by law enforcement.

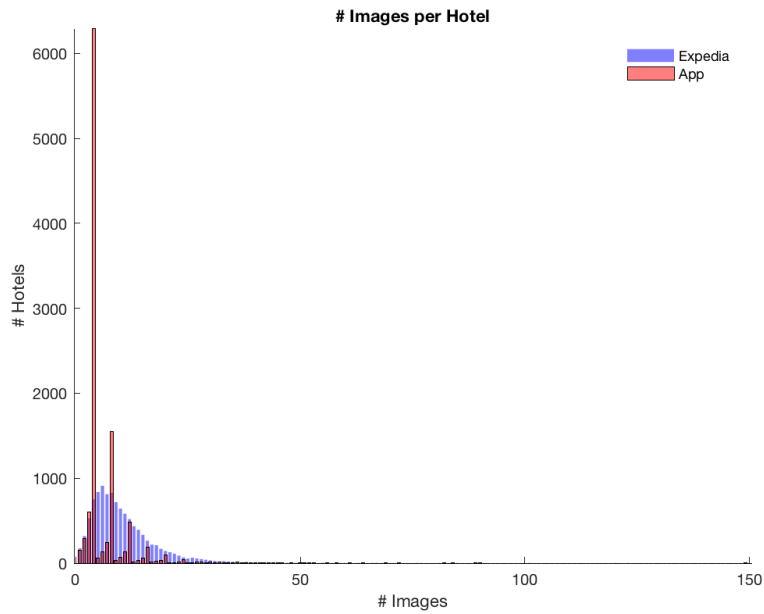


Figure 2.3: The number of images per hotel for both TraffickCam and Expedia.

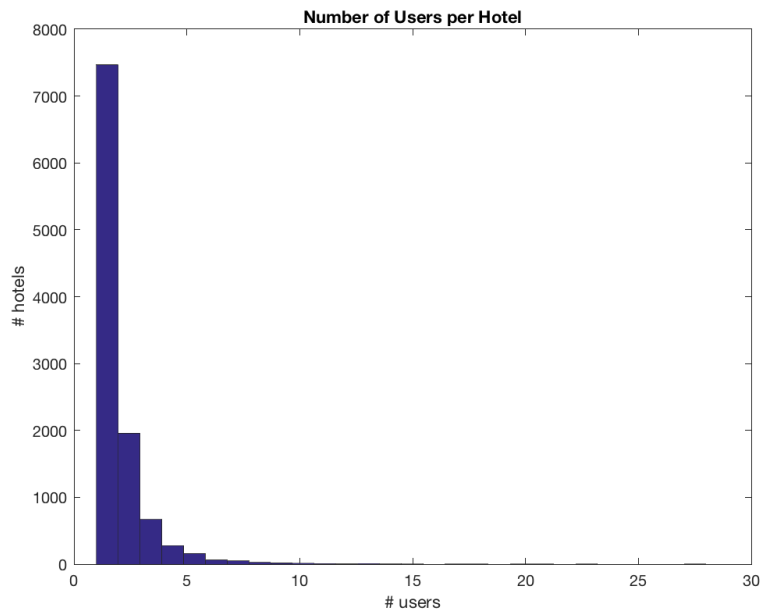


Figure 2.4: This histogram shows that there are largely only a few users at a particular hotel, with a few hotels where a larger number of users have submitted photos.

Chapter 3

Feature Matching Approaches

The approaches to scene recognition that we explore in this work are based on the similarity of descriptors, small numerical representations of either locations in an image (‘local feature descriptors’) or the entire image (‘image descriptors’). Local feature descriptors are either extracted densely (evenly sampled across the image at different scales) or at keypoint locations. There are different approaches to finding keypoints in an image, but the gist of each approach is to find locations in the image with high texture and extract feature descriptions for the regions of the image about those locations. These features can then either be matched directly to each other using a nearest neighbor search, or by mapping the descriptors to an even smaller representation, for example by creating dictionaries of similar features and describing those features with the same ‘word’. Here, we discuss approaches to scene recognition using both local feature descriptors and image descriptors, and different types of matching routines.

3.1 k-Nearest Neighbor Matching with SIFT Features

Local feature descriptors are small numerical representations of small parts of an image. In evaluating local feature descriptors, we focus on David Lowe’s widely used Scale Invariant Feature Transform (SIFT) features [15], and specifically the VLFeat implementation of Lowe’s algorithm [18]. SIFT features are designed to produce similar features (in descriptor space) regardless of the scale or orientation of the image region being described. This is particularly important in the context of matching images of a victim of sex trafficking in a hotel room to a database of hotel room images, as the features (such as the headboard,

curtains, carpet, etc.) may be in very different configurations at different scales between the different datasets.

To evaluate the performance of SIFT features in matching images from hotel rooms, we follow this methodology: given a query image, we first extract SIFT features using the MATLAB implementation [18]. For each of those features, we find the k -nearest neighbors in the set of features extracted from the database of images using the VLFeat’s MATLAB implementation of FLANN’s KD-Tree Forests [18]. Each nearest neighbor match between a feature in the query image and a feature in a database image is a “vote” that the query image was taken in the same hotel as the database image. Votes are weighted by their ranking in the nearest neighbor match (e.g., the first nearest neighbor is weighted more heavily than the fifth nearest neighbor) to determine a list of candidate hotels where the query image might have been taken.

This voting scheme per feature is based off of [21], which addressed the problem of outdoor scene recognition on Google Street View images.

3.2 k-Nearest Neighbor Matching with Learned Features

We compare the performance of SIFT feature matching with feature representations learned from an existing deep convolutional neural network (CNN) architecture [13]. We use a publicly-available, pre-trained model, which we call *Places*, trained on the Places Database [22] for scene recognition from 205 categories (e.g., airplane cabin, hotel room, shed). In this CNN architecture, features are extracted from images in a layered, feed-forward manner. Initial layers of the architecture consist of convolutions, local response normalization, local pooling, dropout layers, and rectified linear (ReLU) activation units. The top layers of the network are four fully connected layers ‘fc6’, ‘fc7’, ‘fc8’, and the final output layer ‘prob’ that represents a categorical probability distribution. The dimensionality of these top layers in Places are 4096, 4096, 205, and 205 respectively. We perform feature extraction using Caffe [10], an open source deep learning framework.

The advantage of this nearest neighbor voting approach is that we search for feature matches over our entire local feature descriptor space. This means that if there is a small crayon mark on the wall in a query image and a matching mark on the wall in another image. In approaches that quantize the local features into a smaller descriptor, those identifying features may be ignored. We could even explore weighting these exact matches proportionally based on the distance between features, as opposed to the default approach in SIFT feature matching proposed by Lowe that checks whether two features are 20% closer to each other in feature space than the next closest match [15]. We do not, however, explore these types of metrics which might further improve the performance of this exact matching approach, as we are looking to implement a real world system and the time constraints of this exact feature matching make this approach to search infeasible.

3.3 Bag of Words Matching with SIFT Features

Matching image features to image features can be extremely computationally expensive. SIFT features are 128-dimensional, and even the non-dense SIFT implementation we use extracts around 1000 features per 480x640 image. Solving for the nearest neighbor of each feature in that 128 dimensional space, as we do in the voting scheme described above, is computationally infeasible for even image databases of a few thousand images. One option to make the search more scalable is to represent each feature not as a 128 element representation, but rather as a single ‘word’ from a dictionary of possible words – an approach known as ‘bag of words.’

To compute a dictionary of possible words, we compute SIFT features from every image in a dataset comprised of 20,000 TraffickCam images, 5,360 indoor images from the Places dataset [22] and 4,250 images from the CalTech Pedestrian dataset [6]. We include the CalTech images in order to learn a more diverse dictionary of words. We then use the SciPy implementation of the k-means clustering algorithm to solve for 5,000 cluster centers, which will be the words in our dictionary [11]. We have experimented with other dictionary sizes, but find that too many words result in words that are too specific, at a loss of robustness to changes in scale and orientation between features. Additionally, larger dictionary sizes take significantly longer for the k-means clustering to converge, without any gains in performance.

To test the performance of this approach, we first extract the SIFT features from both the entire dataset and the query image. Each 128-dimensional feature is then mapped to its nearest cluster center, or word. For an entire image, we can then compute the frequency of each of the 5,000 words in our dictionary for that feature – a ‘bag of words.’ Then, instead of matching each of the 128 dimensional SIFT features (around 1000 per image), we solve for which of 5,000-dimensional bags of words from the database of images is closest to the 5,000 dimensional bag of words from the query.

3.4 Vector of Locally Aggregated Descriptors (VLAD) Matching

Current state of the art approaches to outdoor scene localization are based on vectors of locally aggregated descriptors (VLAD) [9, 1]. VLAD features are similar to bag of words with regards to image retrieval and scene localization, both starting with vector quantization of a large number of local descriptors (like SIFT features) into a smaller representation. Where bag of words encodes the number of features assigned to each cluster center, VLAD instead encodes the distance from the cluster center. The feature size (around 4000 dimensions per image) allows VLAD features to scale well for large scale matching, and the encoding of distance from the cluster center allows for some improvement in the performance of this descriptor for recognition.

We specifically implement a variation of VLAD features called NetVLAD. These features are the product of a convolutional neural network with a ‘VLAD’ layer that aggregates features extracted from the *conv5* convolutional layer. The NetVLAD authors train this network on a large number of Google Street View images for the purpose of outdoor scene recognition [1]. As we show in Chapter 4 Section 4.2, training a model specifically for the task of scene recognition produces better results than using hand tuned features such as SIFT.

We use the author’s MATLAB implementation of NetVLAD, along with the trained model recommended and provided to us by the authors (‘VGG-16 + NetVLAD + whitening, trained on Pittsburgh’), to produce image descriptors and then follow the same procedure for evaluation as described in Section 3.3 of this chapter.

Chapter 4

Experimental Results

We perform two separate sets of experiments on different subsets of the datasets described in Chapter 3 Section 3.1, first matching photos from Expedia to other images from Expedia, and then matching images from the TraffickCam dataset to images from Expedia. Both experiments match images of hotel rooms to other images of hotel rooms. While this problem is likely simpler than matching images of victims of sex trafficking in hotel rooms to images of empty hotel rooms, to date, there is no feasible way to build a labeled dataset of images of victims in hotel rooms. We discuss this challenge and the need for additional work on this front further in Chapter 5.

4.1 Matching Between Expedia Images

The first set of experiments are basic, matching images from the Expedia dataset to other images from the Expedia dataset. For this experiment, the images for both the queries and the database are hand selected from the larger datasets to remove images that are inconsistent with the experiments, e.g., photos of a hotel lobby that are incorrectly labeled as hotel interiors. Within a single hotel, the images are largely consistent in lighting, color and even layout. This is because the Expedia images are largely provided by hotels for advertising purposes, showing off rooms in their best conditions using professional photography and lighting.

Feature Set	Top 1	Top 10	Top 20
SIFT	0.44	0.66	0.69
Places (fc6)	0.32	0.63	0.69
Places (fc7)	0.26	0.54	0.65
Places (fc8)	0.14	0.44	0.52
Places (output)	0.04	0.25	0.31

Table 4.1: Results with baseline feature matching methods. SIFT feature matching performance is better than features extracted from Places in identifying the correct hotel in the single most similar image (Top 1). SIFT features and features extracted from Places (‘fc6’) have similar performance in identifying the correct hotel in the Top 10 and Top 20 most similar images.

In this set of experiments, we compare the performance of matching with SIFT features as described in Chapter 3 Section 3.1 and with the performance of learned feature descriptors as described in Chapter 3 Section 3.2.

For each query image, we follow the methodology detailed in Chapter 3 Section 3.1, and compute the 20 nearest neighbors in the experimental database based on each of the feature type described in the previous section. For each query image, we find which hotel in which each of the 20 nearest neighbor images were captured, and report whether the correct hotel was in the top 1, top 5 and top 20 nearest neighbors.

The results of this experiment are reported in Table 4.1. SIFT feature matching generally has the best performance, identifying an image from the same hotel as the query image as the closest match 44% of the time. SIFT feature matching and matching using the feature extracted from Places layer ‘fc6’ have similar performance when identifying the correct hotel in the top 10 and top 20 closest matches. The places ‘output’ layer has generally poor performance. We show example results for SIFT feature matching in Figure 4.2.

4.2 Matching TraffickCam to Expedia Images

This second set of experiments is more difficult, and hopefully more similar in terms of the discrepancy between the query images and the database of images to the actual problem

of matching photos of victims of sex trafficking to photos of hotel rooms. This discrepancy can be seen in the images in Figure 2.2. In this set of experiments, we attempt to match images from the TraffickCam dataset, which are captured by TraffickCam users on their smartphones with clutter in the room and suboptimal lighting conditions, to images from the more professionally captured Expedia dataset. In order to create the TraffickCam portion of the dataset, we filter for hotel locations where between three and ten users submitted images of the hotel and include all images (both TraffickCam and Expedia) from those hotels. We then train a classifier to label each of those images as hotel room, bathroom or other, using the GoogleNet architecture and 75,000 labeled images from the SUN dataset [20] and the Expedia dataset. This classifier achieves 96.9% accuracy on a test set comprised of SUN and Expedia images (we have not evaluated performance on a test set with TraffickCam images). For these experiments, we evaluate performance in 50 TraffickCam query images from 50 different hotels, and match against a database of 968 Expedia images from 100 hotels including the 50 query hotels.

In this set of experiments, we evaluate SIFT based matching as described in Chapter 3 Section 3.1, bag of words based matching of SIFT features as described in Chapter 3 Section 3.3, and NetVLAD based matching as described in Chapter 3 Section 3.4.

Figure 4.1 shows the metric ‘Any @ N’, for N from 1 to 100, for each matching method. ‘Any @ N’ measures whether there are any instance of the correct answer between 1 and N. This metric is more appropriate for this problem domain than more traditional ‘Recall @ N’, as we care whether we match to any instance of the correct hotel in the top N results, but not whether we match to all instances of that hotel (in fact, we specifically would not want to match a picture of a bed to a picture of a bathroom). The results in Table 4.1 and Figure 4.1 can be compared by observing the ‘Any @ N’ plot for $N = 1, 10, 20$. Overall, the best performance is achieved with NetVLAD features extracted from a convolutional neural network for the explicit purpose of performing well on scene recognition. Figure 4.3 shows example success and failure cases for the NetVLAD feature descriptor. NetVLAD especially outperforms SIFT and BOW matching for low values of N. This is equivalent to finding the correct result on an earlier search page, which is important when considering law enforcement users want to find the correct match as early in the search as possible, as opposed to clicking through many pages of results.

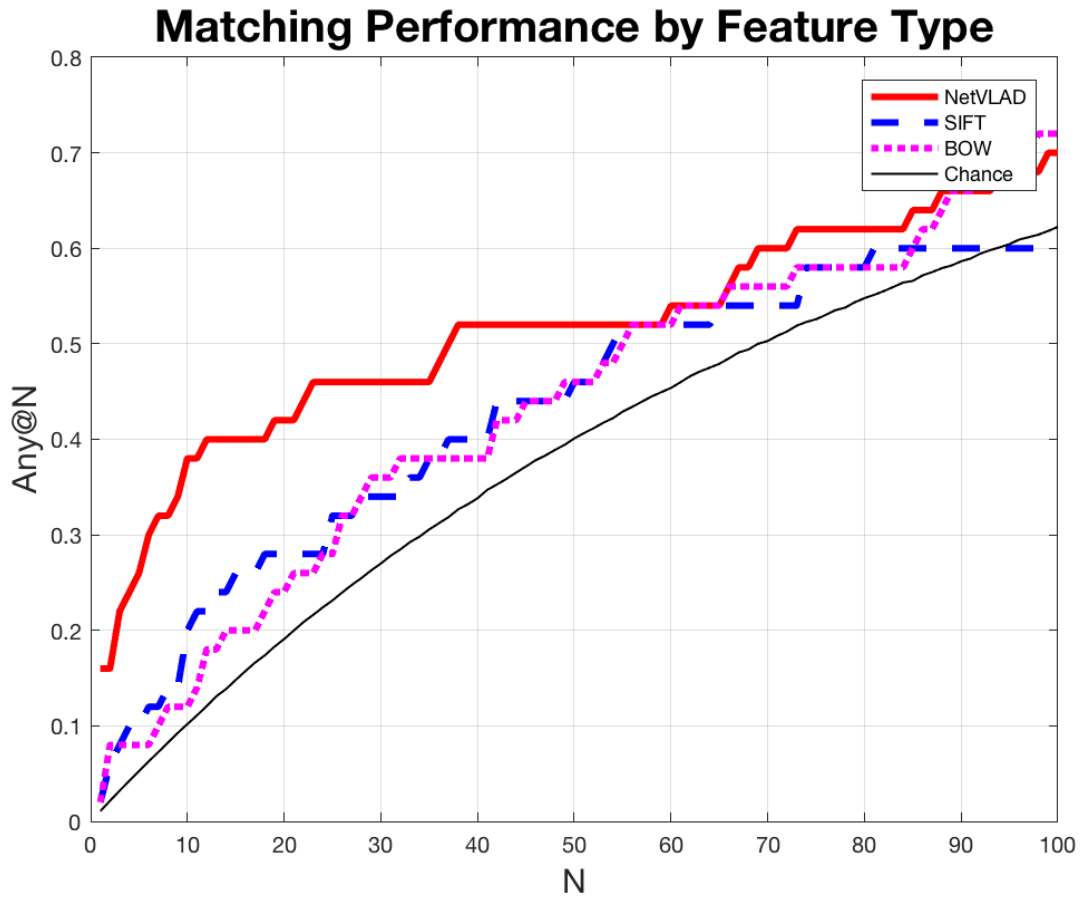
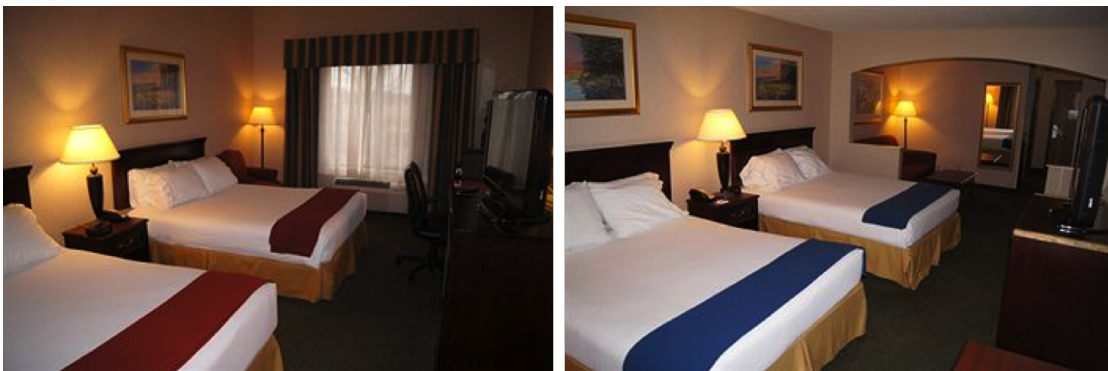


Figure 4.1: We show here matching performance for each of the different feature types in the experiment matching TraffickCam images to Expedia images. We use a variation of the common ‘Recall @ N’ metric, that we call the ‘Any @ N’ metric. ‘Any @ N’ measures whether there are any instance of the correct answer between 1 and N. The best performance is achieved by the NetVLAD whole image descriptor, especially for low values of N, but there is significant room for improvement over all current methods.



(a) A successful matching between images from dramatically different viewpoints.



(b) A successful matching that demonstrates the limitations of our current dataset. These two images are more visually similar than we would ever expect in real world query data.



(c) A failed matching, where SIFT feature matching found visually similar features in the furniture in hotel rooms in two different hotels.

Figure 4.2: The left column shows query images from the Expedia dataset, and the right image shows the image in the same dataset that was found to be the closest match using SIFT features and the matching pipeline described in Chapter 3 Section 3.1. The top two rows show correctly matched pairs, where the query image and result image were taken in the same hotel. The bottom row shows an incorrectly matched pair.



(a) A successful matching between images from different viewpoints and image conditions.



(b) A successful matching between images from very similar viewpoints.



(c) An incorrect, but reasonable matching with similar room configurations.

Figure 4.3: The left column shows query images from TraffickCam, and the right image shows the image which was found to be the closest match in the Expedia dataset, using NetVLAD features and the matching pipeline described in Chapter 3 Section 3.4. The top two rows show correctly matched pairs, where the query image and result image were taken in the same hotel. The bottom row shows an incorrectly matched pair.

Chapter 5

Conclusions and Future Work

This work details a project to find existing Internet imagery and crowd-source the collection of additional imagery of hotel rooms. The dataset creates a resource that can be used in investigations of sex-trafficking because it provides possible locations where photographs of sex-trafficking victims were taken. Our initial results are promising on two different experimental datasets, each including thousands of images either of all the Expedia images from a hotel in a city or of all of the TraffickCam and Expedia images from a random collection of hotels around the United States. Qualitatively, this first experiment is a test on a scale that may itself be useful (if the investigation already knows to focus on a city).

The current experiments are, however, simpler than the real world problem of matching law enforcement provided images of victims of sex trafficking in hotel rooms to images of empty hotel rooms. Building a dataset that is more representative of the types of images that might be used in a law enforcement query (e.g., with large occlusions, sub-optimal lighting, non-professional equipment, etc.), and evaluating different matching methods on that dataset, will be an extremely important area of future work to truly assess the validity of a particular matching routine for this problem domain.

Additionally, the scale of the presented experiments is still significantly smaller than the scale at which the TraffickCam system will be truly useful for law enforcement around the country. While it is possible to create indices of imagery on a local or regional basis that might operate at the scale discussed in this paper (thousands of images), more realistically the system needs to scale to search millions of images at a time. Limiting law enforcement to a single area, when the reality of trafficking is that it often occurs across state lines and regional boundaries, would limit the usefulness of the TraffickCam system.

The first piece of making the system usable at a national scale is to implement a more efficient search over whatever feature we choose. The current search is based on a Python implementation of Whoosh, which was built for text searches. There are alternative search architectures, including Elastic Search, that may be much better suited for our particular search domain and the scale at which it needs to operate.

In addition to improving the search efficiency, there is still room for significant improvement in the accuracy of the system. In Chapter 4 Section 4.2, we show that the best search accuracy is achieved using NetVLAD image descriptors. It is not particularly surprising that a learned feature, trained specifically for the task of scene recognition, outperforms hand crafted features like SIFT on this particular task. However, there is potential for significant improvement over the baseline NetVLAD performance by training the same architecture with a dataset that includes indoor imagery (NetVLAD was trained on outdoor imagery for the purpose of outdoor scene recognition), and is room for exploration of different deep learning architectures that may be even more suited for our particular task.

One additional area of future work that may yield significant improvement in the accuracy of the recognition system is to implement semantic labeling of each scene as a pre-processing step. Existing work, such as [3], demonstrates that convolutional neural networks can be used to provide reasonable pixel-wise classification of an image. Our early experiments in using such a network to label indoor scenes with classes such as ‘bed’, ‘curtains’ and ‘lamp’, have yielded promising results. With these semantic labels, we could then implement a smarter matching routine that only attempts to match features from objects of the same class. This may be particularly important in hotel room matching, where the same carpet or artwork may be present throughout a hotel or hotel chain, while other objects such as furniture might vary.

Finally, there is room to explore whether the existing TraffickCam application captures the best imagery to support the task of recognizing hotel rooms. The application at present asks users to provide images from specific parts of their hotel room in order to maximize the coverage provided of the room. It is possible, however, that there are very specific features that are most important in recognizing a scene – for example, it may be that the curtains or carpet in a room are always the features that are most visible in query images of victims of sex trafficking. In that case, we would want to design the TraffickCam user experience to

provide the images that are most beneficial to the matching problem, such as close ups of the curtains or pictures of the carpet in different lighting conditions.

References

- [1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [2] R. Arandjelović and A. Zisserman. All about VLAD. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [3] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.
- [4] Dave Blair. It takes a movement to defeat a market: Tech-enabled collaboration to combat human trafficking. In *APSA 2014 Annual Meeting Paper*, 2014.
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [6] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009.
- [7] Lisa Fedina. A systematic review and methodological critique of human trafficking prevalence studies. In *Society for Social Work and Research 19th Annual Conference: The Social and Behavioral Importance of Increased Longevity*. Sswr, 2015.
- [8] Kena Fedorschak, Srivatsav Kandala, Kevin C Desouza, and Rashmi Krishnamurthy. Data analytics and human trafficking. In *Advancing the Impact of Design Science: Moving from Theory to Practice*, pages 69–84. Springer, 2014.
- [9] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sanchez, Patrick Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(9):1704–1716, September 2012.
- [10] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

- [11] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.
- [12] Craig A Knoblock and Pedro Szekely. A scalable architecture for extracting, aligning, linking, and visualizing multi-int data. In *SPIE Sensing Technology+ Applications*, pages 949907–949907. International Society for Optics and Photonics, 2015.
- [13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.
- [14] Erin I Kunze. Sex trafficking via the internet: How international agreements address the problem and fail to go far enough. *J. High Tech. L.*, 10:241, 2009.
- [15] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.
- [16] Diana Moise, Denis Shestakov, Gylfi Gudmundsson, and Laurent Amsaleg. Indexing and searching 100m images with map-reduce. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 17–24. ACM, 2013.
- [17] Daniel Ribeiro Silva, Andrew Philpot, Abhishek Sundararajan, Nicole Marie Bryan, and Eduard Hovy. Data integration from open internet sources and network detection to combat underage sex trafficking. In *Proceedings of the 15th Annual International Conference on Digital Government Research*, dg.o '14, pages 86–90, New York, NY, USA, 2014. ACM.
- [18] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008.
- [19] Hao Wang, Congxing Cai, Andrew Philpot, Mark Latonero, Eduard H Hovy, and Donald Metzler. Data integration from open internet sources to combat sex trafficking of minors. In *Proceedings of the 13th Annual International Conference on Digital Government Research*, pages 246–252. ACM, 2012.
- [20] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492. IEEE Computer Society, 2010.
- [21] Amir Roshan Zamir and Mubarak Shah. Accurate image localization based on google maps street view. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

- [22] 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*, 2014.

Indoor Scene Localization, Stylianos, M.S. 2016