ABSTRACT

Title of dissertation: IMAGE GEOLOCALIZATION AND ITS APPLICATION TO MEDIA FORENSICS

Bor-Chun Chen
Doctor of Philosophy, 2019

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

Image geo-localization is an important research problem. In recent years, the IARPA Finder program gathers many researchers to develop the technology to address the geo-localization task. One particularly effective approach is utilizing the large-scale ground-level image and/or overhead imagery with image matching techniques for image geo-localization. In this dissertation, we focus on two different aspects of geo-localization. First, we focus on indoor image and use geo-localization to recognize different business venues. Second, we address the venerability of such a computer vision system and apply geo-localization to solve media forensics problems such as content manipulation and meta-data manipulation.

With the prevalence of social media platforms, media shared on the Internet can reach millions of people in a short time. Sheer amounts of media available on the Internet enable many different computer vision applications. However, at the same time, people can easily share a tampered media for malicious goals such as creating panic or distorting public opinions with little effort.

We first propose an image localization framework for extracting fine-grained location information (i.e. business venues) from images. Our framework utilizes the information available from social media websites such as Instagram and Yelp to extract a set of location-related concepts. Using these concepts with a multi-modal recognition model, we were able to extract location information based on the image content.

Secondly, to make a robust system, we address the metadata tampering detection problem, detecting the discrepancy between the images and its associated metadata such as GPS and timestamp. We propose a multi-task learning model to verify its authenticity by detecting the discrepancy between image content and its metadata. Our model first detects meteorological properties such as weather condition, sun angle, and temperatures from the image content and comparing it with the information from the online weather database. To facilitate the training and evaluating of our model, we create a large-scale outdoor dataset labeled with meteorological properties.

Thirdly, we address the event verification problem by designing a convolutional neural networks configuration specifically target for image localization. The proposed networks utilize the bilinear pooling layer and attention module to extract detail location information from the image content.

Forth, we present a generative model to generate realistic image compositing using adversarial learning, which can be used to further improve the image tampering detection model. Finally, we propose an object-based provenance approach to address the content manipulation problem in media forensics.

IMAGE GEO-LOCALIZATION AND ITS APPLICATION TO MEDIA FORENSICS

by

Bor-Chun Chen

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Professor Larry S. Davis, Chair/Advisor
Professor Rama Chellappa
Professor Tom Goldstein
Professor David W. Jacobs
Dr. Yaser Yacoob

Acknowledgments

First, I would like to thank my advisor, Professor Larry Davis for giving me the chance to work on challenging and interesting projects over the past five years. He provides an open research environment and resources that allow me to explore different interesting topics in the area of computer vision and at the same time always make himself available to provide me help and advice.

I would also like to thank Vlad Morariu, who has been helpful to me as a research mentor during my graduate study. I gain invaluable research experience from discussion with Vlad, as he helps me in polishing the research idea and writing research papers.

I am also grateful to all my committee members, Professor Rama Chellappa, Professor Tom Goldstein, Professor David Jacobs and Doctor Yaser Yacoob for agreeing to serve on my thesis committee and for sparing their invaluable time reviewing the manuscript.

My mentors during my internship at FXPAL (Yanying Chen, Francine Chen), Yahoo! Research (Andrew Kae), and Facebook AI (Sernam Lim) also played an important role in my graduate study. I would like to thank them for the internship opportunities and the research experiences I got during the summer periods.

I would also like to thank my colleagues at the computer vision lab. Thanks to Xianzhi Du, Xiyang Dai, Joe Yue-Hei Ng and Ang Li for their warm welcome when I first joined the lab. I also enjoy lunchtime with Zuxuan Wu, Xintong Han, and Peng Zhou. Discussions with Mahyar Najibi is also really helpful. Not only did I learn a lot from interaction with all the colleagues, but occasional board game sessions with them are also relaxing and joying experiences that help me through my study.

I would also like to thank my mentor from my master program, Winston Hsu, who lead me into the computer vision research world and being a great role model to me.

I owe my deepest thanks to my parents, Hsin-Hsing Chen and Shih-Jung Ma, who have always stood by me and guided me through my life. I would also like to give thanks to my girlfriend, Ruby Teng for her love and support. She has always been patience and understanding during my graduate studies. Finally, I also want to thanks my two cats, Sprinkle and Socks, for their emotional support.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Image localization is important for marketing and recommendation of local business; however, the level of granularity is still a critical issue. Given a consumer photo and its rough GPS information, we are interested in extracting the fine-grained location information, i.e. business venues, of the image. In Chapter 2, we propose a novel framework for business venue recognition. The framework mainly contains three parts. First, business-aware visual concept discovery: we mine a set of concepts that are useful for business venue recognition based on three guidelines including business awareness, visually detectable, and discriminative power. We define concepts that satisfy all of these three criteria as business-aware visual concept. Second, business-aware concept detection by convolutional neural networks (BA-CNN): we propose a new network configuration that can incorporate semantic signals mined from business reviews for extracting semantic concept features from a query image. Third, multi-modal business venue recognition: we extend visually detected concepts to multi-modal feature representations that allow a test image to be associated with business reviews and images from social media for business venue recognition. The experiments results show the visual concepts detected by BA-CNN can achieve up to 22.5% relative improvement for business venue recognition compared to the state-of-the-art convolutional neural network features. Experiments also show that by leveraging multi-modal information from social media we can further boost the performance, especially when the database images belonging to each business venue are scarce.

In order to make a robust system, in Chapter 3, we address the metadata tampering problem. Image content or metadata editing software availability and ease of use has resulted in a high demand for automatic image tamper detection algorithms. Most previous work has focused on detection of tampered image content, whereas we develop techniques to detect metadata tampering in outdoor images using sun altitude angle and other meteorological information like temperature, humidity and weather, which can be observed in most outdoor image scenes. To train and evaluate our technique, we create a large dataset of outdoor images labeled with sun altitude angle and other meteorological data (AMOS+M2), which to our knowledge, is the largest publicly available dataset of its kind. Using this dataset, we train separate regression models for sun altitude angle, temperature and humidity and a classification model for weather to detect any discrepancy between image content and its metadata. Finally, a joint multi-task network for these four features shows a relative improvement of 15.5% compared to each of them individually. We include a detailed analysis for using these networks to detect various types of modification to location and time information in image metadata.

Chapter 4 describe an alternative approach to the metadata tampering detection problem, aiming to verify the authenticity of the metadata associated with the image, using a deep representation learning approach. We propose a deep neural network called Attentive Bilinear Convolutional Neural Networks (AB-CNN) that learns appropriate representation for metadata verification. AB-CNN address several common challenges in verifying a specific type of metadata – event (i.e. time and places), including lack of training data, fine-grained differences between distinct events, and diverse visual content within the same event. Experimental results on three different datasets show that the proposed model can provide a substantial improvement over the baseline method.

In order to further improve the tampering detection algorithm, Chapter 5 describe an algorithm that can be used to generate additional training data with image compositing. Compositing a realistic image is a challenging task and usually requires considerable human supervision using image editing software. We propose a generative adversarial networks (GANs) architecture for automatic image compositing. The proposed model consists of four sub-networks: a transformation network that improves the geometric and color consistency of the composite image, a refinement network that polishes the boundary of the composite image, a discriminator network, and a segmentation network for adversarial training. Experimental results on both synthesized images and real images show that our model, Geometrically and Color Consistent GANs (GCC-GANs), can automatically generate realistic composite images compared to several state-of-the-art methods, and does not require any manual effort.

In Chapter 6, we present an analysis of embeddings extracted from different pre-trained models for content-based image retrieval. Specifically, we study embeddings from image classification and object detection models. We discover that even with additional human annotations such as bounding boxes and segmentation masks, the discriminative power of the embeddings based on modern object detection models is significantly worse than their classification counterparts for the retrieval task. At the same time, our analysis also unearths that object detection model can help retrieval task by acting as a hard attention module for extracting object embeddings that focus on salient region from the convolutional feature map. In order to efficiently extract object embeddings, we introduce a simple guided student-teacher training paradigm for learning discriminative embeddings within the object detection framework. This approach can then be used the retrieve original images from the tampered one in order to identify content manipulation.

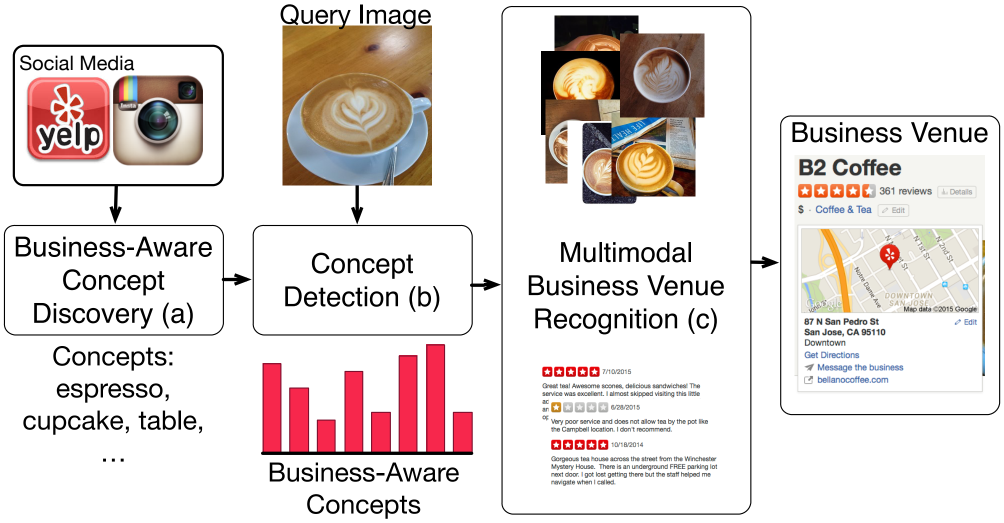
Finally, in Chapter 7, we summarize the dissertation and discuss potential future research direction.

Chapter 2: Business-Aware Visual Concept Discovery from Social Media for Multimodal Business Venue Recognition

2.1 Introduction

Nowadays, there are a sheer amount of images being uploaded to social media sites on the web everyday. Although some of the images contain check-in information that discloses at which business venues they were taken, many of the images do not have such information available. For example, the images uploaded to Flickr or Google Photos only contain GPS information but no check-in information. Even for images which have check-in information, most check-ins are famous travel landmarks while very few of them are local business venues. There arises an interesting research problem: given image content taken in some business venue and its GPS information, we aim to infer which venue the image was taken at.

Recognition of the business venue (e.g. cafe shop, local restaurant) in an image can help many applications for personalization and location-based services/marketing. For instance, it allows personalized promotion based on the business venue a user had visited, or accurate check-in suggestion in social media applications. One might think this is an easy task: since we already have the GPS information, we can just



Great tea! Awesome scones, delicious sandwiches! The service was excellent. I almost skipped visiting this little

Very poor service and does not allow tea by the pot like the Campbell location. I don't recommend.

Gorgeous tea house across the street from the Winchester Mystery House. There is an underground FREE parking lot next door. I got lost getting there but the staff helped me navigate when I called.

Figure 2.1: Given an image uploaded to social media and its rough GPS information, we want to automatically find out the business venue where it was taken. (a) We first mine a list of business-aware visual concepts from social media, (b) use the proposed BA-CNN to detect these business-aware visual concepts from the query image and (c) associate visual concepts with images and business reviews in a geo-tagged database to recognize the business venue.

map it to the GPS information of business venue. However, GPS information is not accurate enough to achieve such fine-grained geo-localization tasks. According to experiments conducted in Maier and Kleiner (2010), modern GPS sensors can have up to 40 meter error, especially in the urban area. Hence, GPS can only help us narrow down the candidates within a nearby area, and we need a more reliable way to recognize the venue.

There are many previous works focusing on geo-localization based on matching visual content. However, most of the works only target on a coarser granularity of location (e.g., city), and they are only applicable for outdoor images while a huge portion of the images on social media websites are indoor images. The major challenge is – indoor images contain less unique visual patterns and many business venues have only a few images associated with them, so it is hard to recognize location in such a fined-grained setting without any high-level semantic descriptions (e.g., coffee cups in the cafe). Some other previous works use text information to infer the user's location. However, these methods cannot deal with the cases when a query image is not associated with any texts and they do not utilize visual information, which can provide useful clues.

By leveraging freely available social media on the Internet, we propose a novel framework to address this challenging problem. As shown in Figure 2.1, our system mainly contains three parts: (1) Business-Aware Visual Concept Discovery: By mining large-scale social media text corpus, we discover a set of business-aware visual concepts that are useful for business venue recognition. (2) Business-Aware Visual Concept Detection: we detect the concepts from images using a novel convolutional neural network configuration (BA-CNN), and (3) Multimodal Business Venue Recognition: we then use Word Vector Model [3] to extend visually detected concepts to word representations and further combine with image content for multimodal venue recognition. Note that the extension of multimodal feature representations only relies on the visual content of a query image without being associated with any texts.

To sum up, the contributions of this paper include: (1) to the best of our knowledge, this is the first work to recognize business venues by using visual content in consumer photos; (2) we develop a systematic framework to automatically mine visually detectable business-aware concepts from reviews of local businesses; (3) we propose a novel CNN configuration to incorporate semantic signals mined from business reviews for training visual concept detector and extracting business-aware semantic features; (4) we extend a visual representation to multimodal feature representations – visual concepts and word vectors – to associate with multiple information sources on the Web for business venue recognition.

2.2 Related Work

Our work is closely related to several research directions. (1) Geo-location prediction: predicting the location information from an image or a short text description (i.e. tweets). (2) Visual concept detection: finding a semantic representation of an image. (3) Convolutional neural networks: learning visual representation based on a deep neural network. In the following section, we will discuss the related works in each area and the differences with our work.

2.2.1 Geo-location prediction

There are many related works for inferring the location from an image. Hays and Efros (2008) is one of the early studies that successfully infer geo-information from a single image. They use a simple data-driven approach to find geo-information based on a large-scale geo-tagged database. However, they only focus on outdoor images with coarse granularity up to city level. Schindler et al. (2007) is another early work on geo-location prediction, which focus on location recognition within a city. They developed an algorithm to select informative low-level features to improve the recognition accuracy in a large-scale setting. While their granularity is smaller, they only focus on street view images within a 20 kilometer range. In Friedland et al. (2010), they use multimodal information to infer the geo-information of a video, but they only focus on city-scale granularity by using low-level feature such as SIFT features [7]. In Fang et al. (2013), they tried to find discriminative image patches for city-level geo-location prediction. In Lin et al. (2015), they use aerial images to help geo-location prediction. While they can achieve a finer granularity, the technique can only apply to images of outdoor buildings. There are also many works that focus on landmark recognition [10] [11], which is highly related to geo-location predication. However, these works relay on distinct low level visual patterns to recognize the landmarks. Note that in [12], they also use GPS information to assist the retrieval task, which is similar to our setting, but they only focus on landmark recognition.

Our work is different from the aforementioned works in many different aspects. (1) We focus on fine-grained business venue recognition, while most previous works only address city-level granularity. (2) We focus on consumer photos which contain both indoor and outdoor images, while most previous works can only deal with outdoor images. (3) We derive a semantic representation from the image content, which can be used to match the text information in the reviews of business venues available in a multimodal database.

There are also many works focusing on geo-location prediction based on texts in the social media (i.e. tweets): Chen et al. (2013) Chen et al. (2014) Hulden et al. (2015) DeLozier et al. (2015). However, text information is not always available and there might not be location-related information available in the texts. Therefore, texts and images can be viewed as complementary sources for geo-location prediction. In this work, we focus on the case where only an image is available as the query for business venue recognition.

2.2.2 Visual concept detection

Our work is also related to the research of visual concept detection. There are many previous works that address generic concepts discovery [17] [18]. However, these concepts are not mined for the purpose of business venue recognition, and therefore, as shown later in the experiments, do not perform well compared to our business-aware visual concepts.

Chen et al. (2014) propose to mine semantic concepts from event description for event detection. Ye et al. (2015) further improve the concept definition by mining concepts from "WikiHow." Compared to these works, we have the following advantage: (1) We consider the discriminative power in terms of business categories while they define a separate set of concepts for each event. (2) We use the features learnt by CNN rather than hand crafted. The concept features in our work are further constrained by the labels of business venues, which incorporate the correlations of concepts associated with the same business venues. (3) We further represent each detected concept as a meaningful word vector that are learned by large-scale review corpus.

2.2.3 Convolutional neural networks

Convolutional neural networks have shown superior performance in many computer vision tasks [21]. Therefore, we adopt it for our visual concept detection. Our CNN configuration is developed based on the one in [22], and implemented with open source framework named CAFFE [23]. Different from the original network structure, our configuration is able to extract semantic concepts while maintain discriminative powers for business venue recognition.

2.3 Proposed Method

2.3.1 System overview

Our goal is to recognize the business venue by a single query image. This section introduces the major components of our system (cf. Figure 2.1): (a) Business-Aware Visual Concept Discovery: mining a list of business-aware visual concepts from a business review corpus. (b) Business-Aware Visual Concept Detection: using a novel CNN configuration to detect the semantic concepts from query images. (c) Multimodal Business Venue Recognition: extending visual concepts to multimodal representation for business venue recognition.

2.3.2 Business-Aware Visual Concept Discovery

We follow three guidelines to discover business-aware visual concepts: (1) *Business Awareness*: the relevance with business venues. For example, “earth” is not a good business-aware concept because it might not be commonly used in any of the business venues; on the other hand, "cat" might be a good business-aware concept because it could appears in local pet shops. (2) *Visually Detectable*: the detectability from visual content in an image. For instance, "disease" although usually appears at hospitals, is hard to be detected by image content, and thus not a good visual concept; on the other hand, "medicine" is a good visual concept because it has more consistent visual patterns for detection. (3) *Discriminability*: the discriminative power to distinguish between different business venues. For example, "person" might not have enough discriminability because it appears in general business venues, while "burger" could be a good concept as it appears more frequently in American restaurants. According to these three guidelines, we first introduce the approach of mining many candidate concepts from reviews of local businesses followed by selecting concepts with high accuracy of visual detection and low entropy across business venues. Figure 2.2 shows an overview of our method for business-aware visual concepts discovery.

2.3.2.1 Mining Candidate Concepts

Following the guidelines mentioned above, we first mine the candidate concepts from reviews of local businesses on a social media website (i.e. Yelp) to ensure the property of business awareness. We first classify the business venues by their top-level category in the Yelp business category topology [1] (example categories include restaurants, active life, automotive, etc.) We then gather 3,000 reviews from each

[1] https://www.yelp.com/developers/documentation

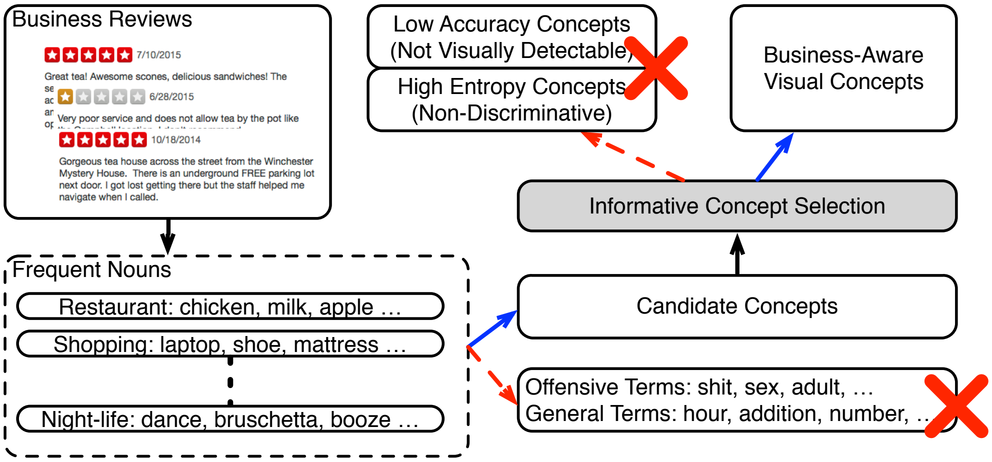

Business Reviews
7/10/2015
Great tea! Awesome scones, delicious sandwiches! The
6/28/2015
Very poor service and does not allow tea by the pot like
10/18/2014
Gorgeous tea house across the street from the Winchester Mystery House. There is an underground FREE parking lot next door. I got lost getting there but the staff helped me navigate when I called.
Low Accuracy Concepts (Not Visually Detectable)
High Entropy Concepts (Non-Discriminative)
Business-Aware Visual Concepts
Informative Concept Selection
Frequent Nouns
Restaurant: chicken, milk, apple …
Shopping: laptop, shoe, mattress …
Night-life: dance, bruschetta, booze …
Candidate Concepts
Offensive Terms: shit, sex, adult, …
General Terms: hour, addition, number, …

Figure 2.2: The overview for business-aware visual concept discovery. We first collect Yelp reviews and find frequent nouns in every business category, and then remove general terms (to every category) and offensive terms (blocked by Instagram) to construct a set of candidate concepts. Finally, we select concepts with visual consistency and low normalized entropy across locations.

business category respectively. From each category, we select 500 frequent nouns based on their document frequency as our candidate concepts. Note that we use NLTK Toolkit [24] to tokenize the words in the reviews and find the part-of-speech tags. We only select the nouns as candidate concepts to ensure the concepts are more visually detectable. There are many overlapping concepts in each category and we find 2,143 concepts overall. In order to ensure the discriminability of the candidate concepts, we remove concepts that appears in more than ten different categories. We also remove concepts that are offensive terms that blocked by Instagram API and result in 1,723 concept candidates. Table 2.1 shows some candidate concepts found in each category.

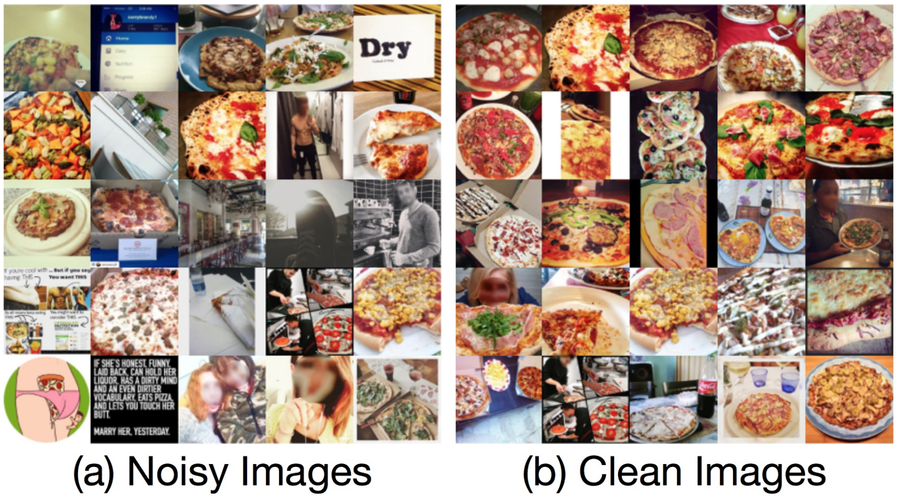
Category	# of Concepts	Example Candidate Concepts
Restaurants	233	chicken, milk, apple, sashimi, onion, tea, chef, pasta, waiter, pizza
Pets	190	doctor, vet, furry, tail, adoption, cage, toy, cat, doggie, salon
Automotive	184	motorcycle, windshield, carpet, auto, girlfriend, stereo, wheel, gas, tank, dealership

Table 2.1: Example candidate concepts in each category mined from reviews of local business.

2.3.2.2 Selecting Informative Concepts

After finding candidate concepts, we need to select useful concepts for business venue recognition from an image. For each concept, we use it as keyword to retrieve 1,000 images from a social media website, i.e. Instagram. Since images downloaded from Instagram are quite noisy, we do two-fold cross validation by using convolutional neural networks (CNN) [22] to select qualified images for learning accurate detectors of visual concepts.

The main idea of two-fold cross validation is – dividing the images into two sets, training a separate concept classifier for each set, and finally using each to verify images in the other set. We select top 250 images from each set based on the classification score for training the concept detectors. Figure 2.3 (a) shows the training data before the cross-validation selection for concept "pizza" while Figure 2.3 (b) shows the training data after cross-validation selection. We can see that the training data after selection are more visually consistent and therefore can achieve better accuracy for concept classification. The experiment in Table 2.2 shows that by cross-validation selection we can achieve up to 48.5% classification accuracy compared to 36.5% by simply using all images as training data. Finally, we remove concepts that have validation accuracy lower than 50% (using hash tag



(a) Noisy Images

(b) Clean Images

Figure 2.3: (a) Images crawled from Instagram by the hash tag “pizza.” (b) Images selected by cross-validation that are more visually consistent and correctly represent the visual concept.

Training Data	All	Random	CRV
Rank-1 Accuracy	36.5%	38.7%	**48.5%**

Table 2.2: Accuracy of concept classifiers trained by all images (All), randomly selected images (Random) and the images selected by cross-validation (CRV). Note that the accuracy involves the concepts that are less visually detectable. After concept selection, CRV can reach 85% accuracy.

as ground-truth) to ensure the visual detectability of concepts.

We then further select the concepts with more discriminative power by computing the cross-location normalized entropy using the following formula:

$$\eta(X^{(c)}) = -\sum_{i=1}^{n(c)} \frac{p(x_i^{(c)}) \log_2(p(x_i^{(c)}))}{\log_2(n^{(c)})}, \tag{2.1}$$

where X is a random variable that denotes the venue distribution of concept c. $\eta(X^{(c)})$ is the normalized entropy for that concept. $n^{(c)}$ is the total number of business venues that have concept c and $p(x_i(c))$ is the probability of the concept

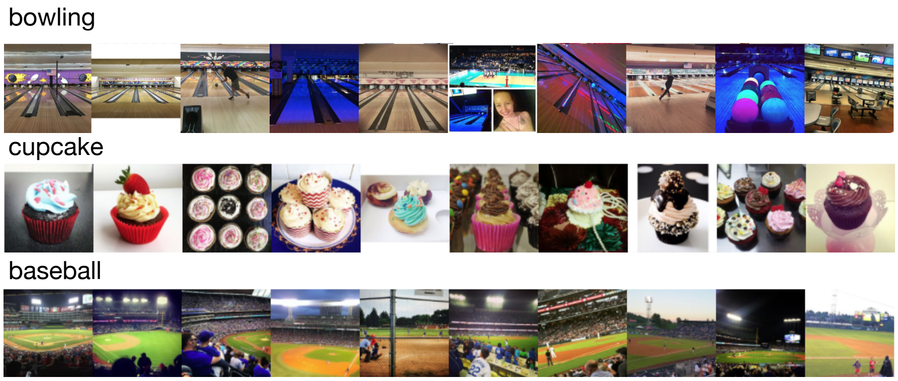


Figure 2.4: Example concepts and corresponding images.

appears in a business venue i. We prepared a dataset from Instagram that contains 250,000 images associated with 1,000 different business venues and computed the normalized entropy for each concept in terms of its distribution over business venues. Finally, the 490 concepts with the lowest entropy value are selected as business-aware visual concepts for business venue recognition. Figure 2.4 shows some example concepts and corresponding images.

2.3.3 Convolutional Neural Networks for Business-Aware Concepts (BA-CNN)

Convolutional Neural Networks have shown promising results in many computer vision related problems. Here we adopt the state-of-the-art visual features learned by CNN [22] as a baseline for business venue recognition. Note that because of (1) scalability: too many business venues and (2) sparsity: only a few images for most business venues (cf. Figure 6), we cannot directly train the classifiers to distinguish different business venues. Instead, we learn the features supervised by

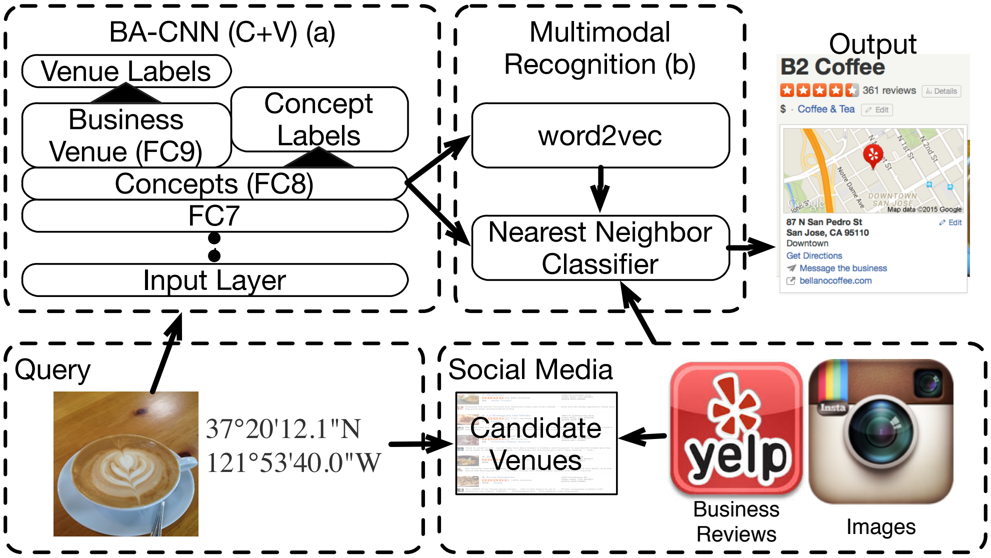


Figure 2.5: System framework for multimodal business venue recognition. Given an query image, we first find a list of candidate venues from social media using GPS, and detect business-aware concepts from image content using BA-CNN (C+V). We then use a Word Vector model to generate the text representation. The visual concept scores and text representation of the query image are then matched against those extracted from the reviews and images in the database. The business venue associated with the best-matched images and reviews is returned as the most likely business venue.

different types of labels at the output layer of an CNN, and use the activations from the last fully-connected layer (FC7) before the output layer as the features to represent an image. The types of labels could be: general concepts used in ImageNet (ImageNet-CNN), business-aware concepts (BA-CNN (C)) and a subset of business venues (BA-CNN (V)). The comparisons of different types of labels are presented in the experiments later. Finally, we apply nearest neighbor classifier based on the CNN features of an query image and database images. The business venue associated with the most similar database image is output as the predicted business venue. Note that the GPS of the query image is used to narrow down the candidate business venues. The impact from the number of candidates is discussed in the

Experiments section.

However, simply use CNN features may suffer from several problems. For ImageNet-CNN (i.e. a network trained on ImageNet labels), the concepts are predefined and not relevant to local businesses; for BA-CNN (C) the discriminability only lies in separating different business-aware concepts rather than business venues; finally, BA-CNN (V) the business venues are limited to the venues comprising more training images and thus cannot cover general business venues. Furthermore, the common problem of CNN features is – they do not have semantic meaning, which is a key property to associate with other data domains.

To address these issues, we propose a new CNN configuration (BA-CNN (C+V)) to detect business-aware concepts for business venue recognition. As shown in Figure 2.5 (a), instead of using FC7 for recognition, we let layer (FC8) supervised by business-aware concept labels and add another layer (FC9) on top of the concept layer supervised by a subset of business venue labels. This way, we can extract features from FC8, where each dimension corresponds to a business-aware visual concept, and has the discriminative power to separate different business venues. In our experiments, BA-CNN (C+V) is demonstrated with a higher recognition accuracy compared to the other CNN features extracted from images. Moreover, it is able to associate multimodal data (e.g., text and images) for recognition since the features extracted by BA-CNN (C+V) are the responses of semantically describable concepts.

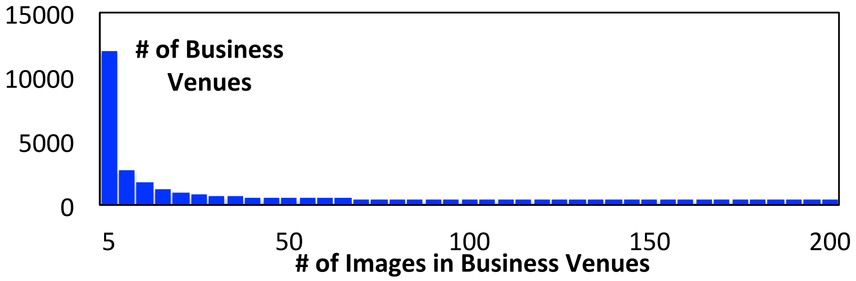


Figure 2.6: The number of images in each business venue sampled from social media ($> 50\%$ venues have < 5 images).

2.3.4 Multimodal Business Venue Recognition

Once we have the concept representation detected by BA-CNN, we can use it for business venue recognition. However, we want to further improve the recognition accuracy by extending image content to multimodal representations – visual concepts and text representation, to utilize the text information, i.e. business review, of the business venues available on the social media. Figure 2.5 shows our system framework for multimodal business venue recognition.

We first use review text of local businesses (e.g. Yelp reviews) to train word vector model [3] that can convert each word into a 500-dimensional vector representation. For each query image, we use the top-5 visual concepts detected from the query image as concept words and average the word vector representation of the top-5 concepts to represent another modality of the image. As shown in Figure 2.5 (b), visual concept representation and word vector representation are then fused together to form the final representation. Here we simply use early fusion (i.e. concatenate the 490 dimensional concept representation and 500 dimensional word vector representation together to form a 990 dimensional vector) to combine two modalities. Similarly, the images and reviews associated to business venues in the databases are also represented as visual concepts and word vectors, respectively. Finally, we use a nearest neighbor classifier with L2 distance based on the multimodal representation to determine the most likely business venue.

2.4 Experiments

2.4.1 Data Collection and Experimental Settings

For our experiments, we need images and reviews related to business venues. We use the public data, Yelp Challenge Dataset [2], which contains information and reviews of 61,184 business venues in ten different cities from Yelp for this purpose. We then map the venues to the Instagram checkin based on GPS information and venue name. 22,763 venues were found on Instagram. We collect up to 1,000 images for each venue. The distribution of images over venues is shown in Figure 2.6. Note that more than a half of the venues have fewer then five images. We take 250 images from each of 1,000 different venues as training data to train the BA-CNN and to compute the normalized entropy in each concept. We than take the other venues with more than eleven images as our evaluation set. In total, 7,699 venues are used for evaluation. For each venue, we randomly select one image as query image. The remaining 10 images together with 20 Yelp reviews of the venue construct a geo-tagged database, where the visual concepts (image) and the word vector (reviews)

[2] http://www.yelp.com/dataset_challenge

are used to represent the associated business venue. During the recognition, we use GPS information from the query image to narrow down candidate venues to two to ten neighboring venues. We use rank-1 accuracy as our evaluation metric.

2.4.2 Improvements by BA-CNN

We compare **BA-CNN** with several baselines and different settings: (1) **ImageNet-CNN (FC8)** [17]: we use responses of general concepts (FC8) from CNN trained on ILSVRC 2012 data as a baseline feature. (2) **ImageNet-CNN (FC7)** [21]: we use CNN trained on ILSVRC 2012 to extract features (FC7) for business venues recognition. (3) **BA-CNN (C)**: we use CNN trained on Instagram images labeled with 490 business-aware visual concepts to extract features from FC7. For each of the 490 concepts, we further collect 4,000 images from Instagram and use 2,000 images with higher classification scores as training data, in total around one million images are used for training. (4) **BA-CNN (V)**: we use 250,000 images from 1,000 different business venues as training data to train CNN and extract features from FC7. (5) **BA-CNN (C+V)**: we use the configuration in Figure 2.5 (a) to extract the business-aware concepts for recognition.

As shown in Figure 2.7, for every method the accuracy drops when the number of neighborhood venues increase because the task becomes more difficult. However, BA-CNN (C+V) can achieve up to 77.5% accuracy when there are two candidates and still maintain around 45% accuracy when the candidate numbers increase to ten; overall, the performance is the best against the other baselines.

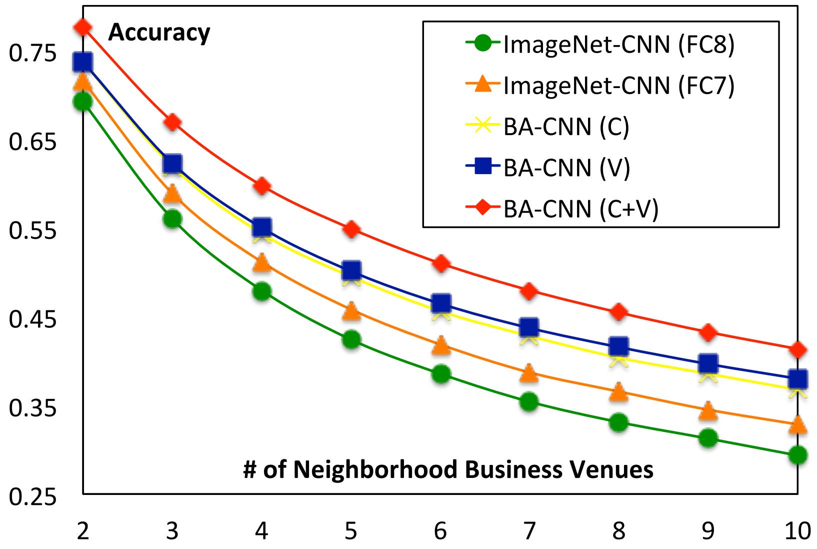


Figure 2.7: Recognition accuracy as different numbers of neighboring business venues are considered as candidates. When there are more business venues nearby, the performance will drop because the task becomes harder. BA-CNN (C+V) outperforms all other baseline consistently.

ImageNet-CNN performs much worse than BA-CNN and the relevant approaches because the concepts in ImageNet are generic concepts without considering business awareness and discriminative information between business venues. BA-CNN (C) and BA-CNN (V) have similar performance but BA-CNN (C+V) outperforms both methods because it utilizes both the concept and venue label information in a hybrid structure. Also, BA-CNN (C+V) can take advantage of the semantic representation and be used for multimodal recognition as shown in the following section.

2.4.3 Results of Multimodal Business Venue Recognition

We use the word vector model to convert the visual concepts detected from the query image and the reviews of each business venue in the database as a vector of text representation. Table 2.3 exhibits the accuracy of business venue recognition by matching the text representations only, that is, no database images are used. **WordVec (Google News)** shows the performance of the model trained with Google News dataset (about 100 billion words) and **WordVec (Business-Aware)** indicates the model trained with Yelp reviews (about 0.2 billion words). **Random Guess** is the accuracy of randomly picking one of the candidate venues. We can see both methods outperform random guessing significantly (more than 115% relative improvement), which suggests that the concepts generate from BA-CNN (C+V) indeed have semantic meaning and highly relevant to what might appear in reviews of local business. WordVec (Business-Aware) performs slightly better than WordVec (Google News) that again shows the importance of business-awareness in the application of business venue recognition.

When combining BA-CNN (C+V) with Word Vectors, we can further improve the recognition accuracy, demonstrating the complimentary nature of the image and text information. It is worth noticing that the multimodal recognition only requires an query image without any text because the proposed image representation, business-aware visual concepts, can be used directly when text representation is available.

The multimodal representation is particularly important for the image sparsity problem in the database of business venues. As shown in Figure 2.6, many of the business venues contains fewer than five images on Social Media website. Therefore, we also evaluate our method with different number of images (range from one to ten images) for each business venue. In Figure 2.8, “WordVec” indicates the accuracy of matching query image and database reviews when no database images are available. As the number of database images in the business venues decreases, the recognition accuracy by image representations drops. “ImageNet-CNN (FC7)” only outperforms “WordVec” when there are more than three images of each venues in the database. The accuracy is obviously boosted by further considering database reviews (“BA-CNN (C+V)” vs. “BA-CNN (C+V) + WordVec”) when few images are available, suggesting the proposed multimodal recognition method have advantages to tackle the image sparsity issue. In social media, the associations between images and venues are mainly based on user checkins. However, because of the heavy tail and power law behavior of checkins per venue [25], only a few famous venues feature a large number of checkin images, while general business venues have only few checkin images. In consideration of this problem, our approach poses a new opportunity to push the generality of automatic recognition to common business venues.

2.5 Conclusion

We propose a novel framework for business venue recognition. We first mine business-aware visual concepts from reviews of local business, and then incorporate business-aware concepts with convolutional neural networks for representing images

Method	Acc.@2	Acc.@5
Random Guess	50.0%	20.0%
WordVec (Google News)	65.8%	39.1%
WordVec (Business-Aware)	69.1%	42.3%
BA-CNN (C+V) + WordVec	**78.5%**	**56.1%**

Table 2.3: The recognition accuracy with 2 and 5 candidate venues. Simply using text representation obviously outperforms random guess, suggesting the concepts extracted from BA-CNN (C+V) indeed have semantic meaning. WordVec (Business-Aware) surpasses WordVec (Google News) demonstrating the importance of business awareness. BA-CNN (C+V) + WordVec can reach the best accuracy.

as response of visual concepts. The semantics of visual concepts can be further represented by text representation. We propose to use multimodal representation for business venue recognition and the experiments show its superiority against the single modal approaches and the state-of-the-art visual features, especially when there are insufficient images to represent the venues. In the future, we will seek the opportunity to associate more data domains, e.g., company profiles, purchase logs. Moreover, we will investigate the other metadata that can replace GPS to narrow down candidate venues, e.g., time, social network.

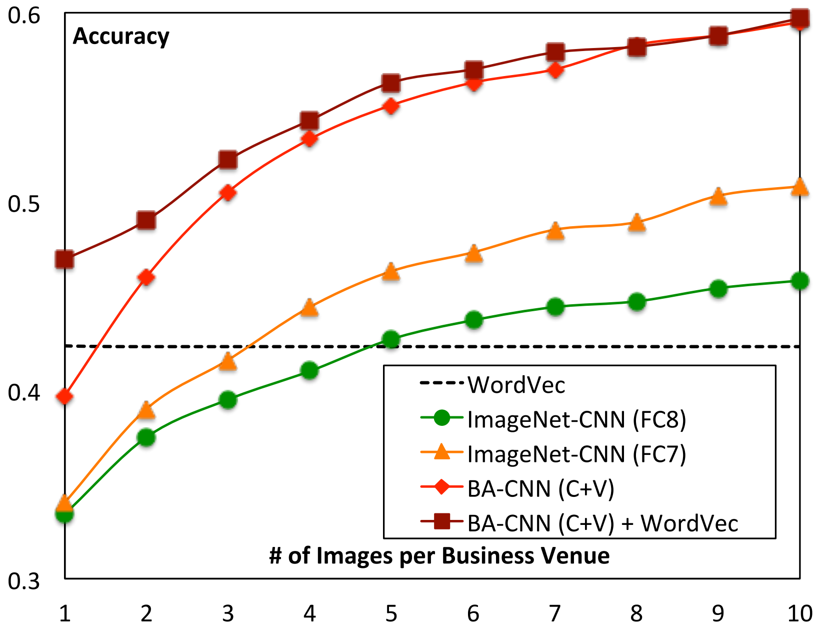


Figure 2.8: The accuracy with different number of images for each business venue. Image sparsity decreases the accuracy of the models using image representation, while text representation is stable, and multiple modalities (BA-CNN (C+V) + WordVec) can improve more in such cases.

Chapter 3: Detection of Metadata Tampering through Discrepancy between Image Content and Metadata using Multi-task Deep Learning

3.1 Introduction

Tampered image metadata is frequently encountered in image forensics. Ease of metadata access and modification using simple EXIF tools has resulted in tampered images that are difficult to detect, except in very special cases or after rigorous expert investigations. Our goal in this paper is to automate this process and reduce the effort required by experts.

One of the areas where image metadata authenticity is very important is legal cases where an image is shown as evidence of a certain activity at a certain time. The time-stamp of the image cannot be trusted just on its own as it is easily modified. It needs to be corroborated by some additional information in the image if available. For example, in the Duke Lacrosse case [26], the timestamp of one of the images matched with the timestamp of one of the player's watch. We develop automatic techniques to perform similar types of analysis in outdoor images using meteorological information.

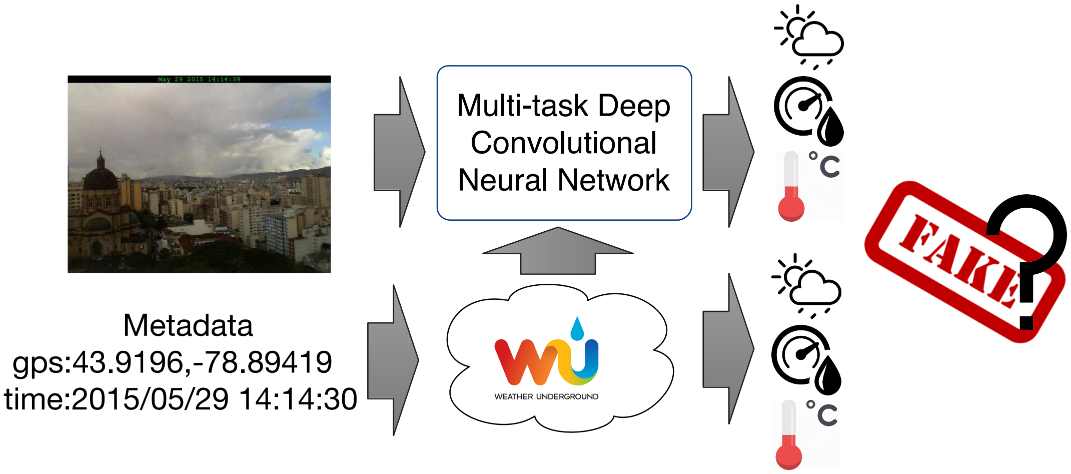


Figure 3.1: Given an image, we first detect information such as sun altitude angle, temperature, humidity and weather conditions using multi-task deep learning. We then compare the inferred properties to the same information collected from the Internet based on image metadata to detect if there is any tampering.

We focus on image location and timestamp tamper detection, as these two are the most important factors in the image metadata. Existing research has focused on checking the validity of the location information by matching image content against a large-scale image database such as Google street view images using content-based image retrieval techniques. However, this only works well with very few locations having distinct features such as tourist landmarks.

Although it is hard to directly infer location and time from the image content, recent research has shown that advances in machine learning have enabled reasonably accurate prediction of meteorological information directly from image content [27, 28, 29, 30]. Therefore, we utilize the sun altitude angle and other historical meteorological information such as temperature, humidity and weather — all available on the web — to detect image metadata tampering. Our goal is to infer meteorological properties separately, directly from image content and then compare them to the same properties obtained from historical weather databases at the time and location specified in the image metadata. We expect that, unless the image metadata was carefully tampered with to ensure consistency with weather patterns, metadata tampering will lead to inconsistencies that can be detected by our proposed algorithm.

To train and evaluate our approach, we first collect a large-scale dataset (AMOS+M2) with images, metadata (i.e., timestamps and GPS location), as well as sun altitude angle and meteorological information based on the already existing AMOS [31] database and the Weather Underground Internet API [32]. We then use AMOS+M2 to learn different convolutional models for prediction. In order to utilize the correlation between different sources of information, we further propose a joint model based on multi-task learning, which predicts all of the features simultaneously.

While there has been some work in this area, our novelty lies in the fact that our test and training data comes from different web cameras, and our research includes the results of applying these models to image forensics. Also, by combining different networks using multi-task learning, we are able to further improve the prediction accuracy. Figure 6.1 shows the overview of our system.

The main contributions of this paper include: (1) analyzing the use of sun altitude angle and meteorological information for image content vs. metadata discrepancy detection; (2) exploiting the benefit of multi-task learning on meteorological information and sun angle prediction; (3) constructing a large-scale dataset called AMOS+M2 containing more than 500,000 outdoor images labeled with the above mentioned information and the metadata.

3.2 Related Works

There has been a large amount of research in the field of digital image forensics. Sencar [33] provide a survey of the different available digital image forensics techniques. The survey includes methods based on image source identification, synthetic image identification, and detection of image tampering. Most of the tampering detection techniques perform statistical analysis of the different kinds of variations in the observed signals after tampering.

Although there have been many successes in detecting tampering from image content, existing techniques generally do not deal with image metadata tampering. Kakar [34] is one of the few that have addressed this problem. However, instead of using only sun angle for detection, we combine other meteorological information available on the Internet and apply multi-task deep learning to further improve accuracy.

Other related works have focused on prediction of sun angle or other meteorological information: Lalonde [35] use mathematical models based on sun illumination, shadow length and direction and shading of vertical surfaces to estimate the sun position and illumination, and others have also investigated similar approaches [27, 29, 30, 36]. Recently, Volokitin [28] applied deep convolutional neural networks for temperature and time prediction. However, none of these methods utilize different meteorological information with multi-task deep learning. Some of them only train and test on images from the same webcam. Our goal is to learn a general model that can be applied to any outdoor image, captured by any camera, at any location or time for metadata tampering detection.

3.3 Sun angle and meteorological information prediction

We use convolutional neural network (CNN) models to predict sun angle and meteorological information. We experiment with two variants of convolutional models for our prediction tasks: AlexNet [22] and ResNet-50 [37] . AlexNet contains five convolutional layers followed by three fully connected layers, while ResNet-50 contains 49 convolutional layers with residual connections followed by one average pooling layer. We use AlexNet to experiment with different loss functions (mean squared and mean absolute losses) due to the advantage of its training speed and use ResNet-50 to train our final model to obtain better prediction results.

3.3.1 CNN for temperature, humidity, and sun angle regression

To use CNN for regression tasks, we first replace the last layer of the CNN with a single output using a distance based loss function. Since the outputs of our regression models should always lie in certain ranges (e.g. zero to ninety degrees for sun angle), we use a sigmoid or an extra ReLU-like nonlinear layer to clip the output from both sides before the final loss layer; but they improve performance only in some cases whereas decrease performance in others. We also weight the training loss based on the probability distribution of the ground truth labels and call these the weighted regression models. This helps to give more importance to the examples that are less common in the training set and tries to solve the problem that the dataset is not uniformly distributed. Finally, we train the network with our AMOS+M2 dataset.

3.3.2 CNN for weather classification

For weather classification, we train a classification CNN with our AMOS+M2 dataset. We first separate our training data into four different classes: sunny, cloudy, rainy, and snowy. Since our training set is highly unbalanced, as sunny and cloudy images together take around 85% of the training set, directly training the network would cause the model to be biased toward sunny and cloudy. To address this issue, we apply data oversampling with augmentation: for each image class, we first oversample the images to make each class have roughly the same size, and then we apply data augmentation to each oversampled image by first randomly resizing and keeping the smallest side of the image between 256 to 512 pixels. We then randomly crop the image down to 227×227 and randomly apply a left-right flip to the image. Finally, we adopt the softmax cross entropy loss function to optimize the network parameters. In order to reduce the training time, we initialize the weights of our network to a model pretrained on ImageNet dataset.

3.3.3 CNN with joint multi-task learning

Since all of the meteorological information we use is correlated, it is natural to wonder if one model can benefit from the others. Therefore, we use multi-task learning to learn a joint model that can predict all the meteorological information at the same time. This is achieved by weight sharing on all the regression and classification networks with a joint loss function. We adopt the same network architecture, ResNet-50, for all tasks so that we can share the weights crossing all four tasks. Since the four different tasks have different output ranges, we first normalize each output to zero mean and one standard deviation so that each loss function will be the same scale. Let $X = [x_1, ..., x_7]$ be the output of our joint network, and $Y = [y_1, ..., y_7]$ the value of the meteorological information, where $(y_1, ..., y_4)$ is a one-hot encoding vector of weather condition, and y_5, y_6, y_7 represent sun altitude angle, temperature, and humidity respectively. We minimize the following joint loss function:

$$L(X, Y) = -\sum_{i=1}^{4} \log y_i p(x_i) + \sum_{i=5}^{7} ||x_i - \frac{(y_i - \mu_i)}{\sigma_i}||^2, \tag{3.1}$$

where, $\mu_5, \mu_6, \mu_7, \sigma_5, \sigma_6, \sigma_7$ represent the mean and standard deviation of sun altitude angle, temperature, and humidity in the training set. $p(x_i)$ represents the probability of the i_{th} class being the correct weather computed by the softmax function. We train this joint model with an initial learning rate of 0.0002 and a mini-batch size of 256 images using Adam optimizer [38].

3.4 Metadata and meteorological information outdoor scenes dataset

In order to train our model for metadata tampering detection, we construct a large-scale image dataset called AMOS+M2.

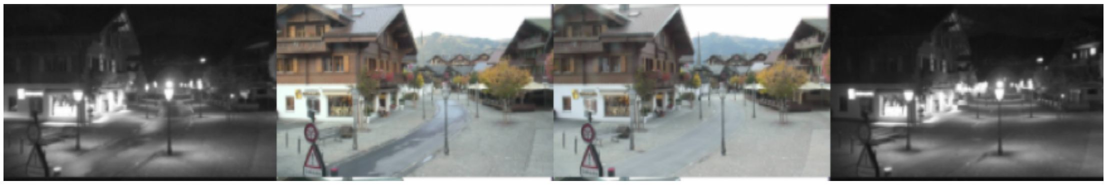


Figure 3.2: Example of boundary images where the sun altitude angle changes from negative to positive. We use these boundary images for manually verifying the camera location. If the camera geographic location is incorrect, the calculated sun altitude angles will be incorrect, and therefore, it is less likely that such day/night boundary can be identified in the image content. So we check boundary images to filter out cameras with incorrect location annotations.

Dataset	# of locations	# of images	Metadata	Meteorological information	Sun angle
Weather Image Dataset [29]	N/A	10K	N	weather	N
Multi-class Weather Image [30]	N/A	20K	N	weather	N
Glasner[39]	10	6K	Y	temperature	N
Time of the Year Dataset [28]	10	23K	Y	temperature	N
AMOS+M2 (Ours)	638	500K	Y	weather, temperature, humidity	Y

Table 3.1: Comparison between AMOS+M2 with other existing datasets. AMOS+M2 contains more images from different locations; with more detailed meteorological information as well as sun altitude angles.

3.4.1 Data collection

We collect images from Archive of Many Outdoor Scenes (AMOS), an archive of images collected from Internet webcams since 2006. Each image in AMOS contains a timestamp and a camera ID, and each camera may contain its location annotated by the AMOS user as well as the IP location of the webcam. Note that the timestamp associated with any image is mostly correct because it is automatically generated by the system, but the location of the camera can be missing or incorrect.

In order to verify the location of the cameras, we first compute the distance between the location derived from the camera IP address and the annotated location

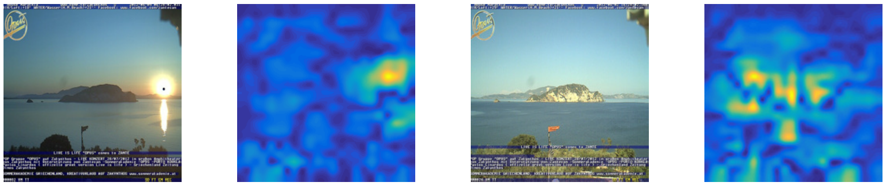


Figure 3.3: Heat maps of absolute difference in output sun altitude angle predictions when small portions of the images are occluded. The two images are from the same webcam at different times. In the first set, we can see that the network gives importance to the sun if it is visible in the image. In the second set the network gives importance to the reflective rock surfaces.

and filter out cameras when this distance is greater than 100 miles. We then compute the sun altitude angle for each image based on the timestamps and the annotated camera location using Pysolar [40] and detect the sunrise and sunset boundary, where sun angle changes between positive and negative numbers. If the location is correct, we should be able to visually see large illumination differences between these boundary images as shown in Figure 3.2. We manually check these boundary images to remove cameras with incorrect GPS locations.

After manual verification, we use the Weather Underground API [32] to collect all the relevant meteorological information including temperature, humidity and weather conditions based on the locations and the timestamps of the images.

3.4.2 Dataset statistics

We obtain 638 cameras from AMOS with verified locations. We randomly select 538 cameras for training and the remaining 100 cameras are used for validation. For each camera in the training set, we randomly select around 1,000 images taken in 2016 to construct a training set of 500,000 images; for each camera in the testing

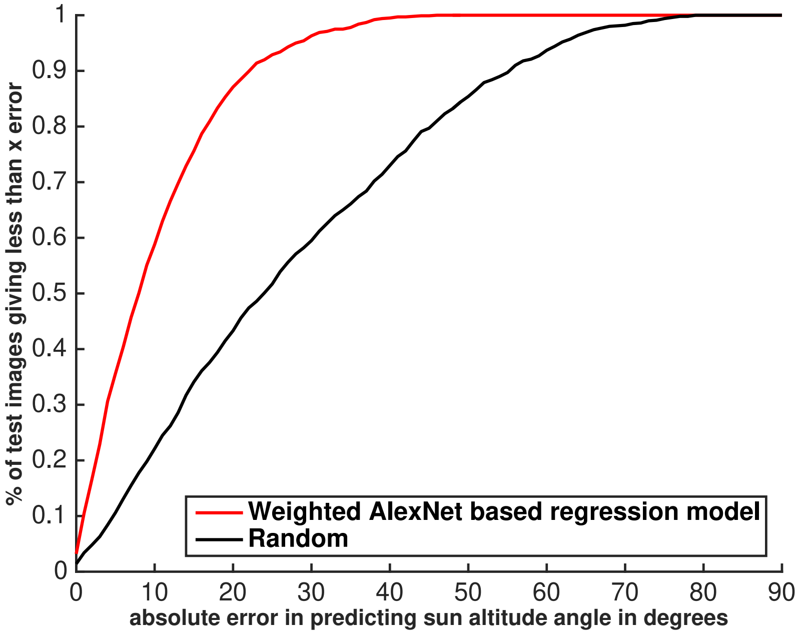


Figure 3.4: The x-axis in the figure is the absolute error in the prediction of sun altitude angle and the y-axis is the percentage of test images giving error less than or equal to the corresponding x value. The higher the area under the curve, the better is the result. For the sun altitude angle test set, the model resulted in 55% of test images with less than or equal to 10error and about 85% of test images with less than or equal to 20error.

set, we randomly select 10 images taken in 2016 to construct a test set of 1,000 images. Table 3.1 shows the dataset statistics compared to related works. Compared to existing datasets, AMOS+M2 contains more images from multiple locations, and more detailed meteorological information, as well as sun altitude angles, enabling us to effectively train our convolutional models.

The AMOS+M2 dataset with the images and corresponding meteorological data and metadata will be made publicly available.

3.5 Experiments on meteorological information and sun altitude angle prediction

3.5.1 Sun altitude angle regression

The performance of an AlexNet based L2 regression model for sun altitude angle is shown in Figure 3.4. The x axis in the figure is the absolute error in the prediction of sun altitude angle and the y axis is the percentage of test images giving error less than or equal to the corresponding x value.

Figure 3.4 shows that almost 55% of the images yield less than 10error and about 85% of images give less than 20error for the weighted regression model. The RMS sun angle prediction error for this model is 13.70. On the other hand, the Resnet based model gives an RMSE of 11.31.

To gain insight into the internal representation of the model, we visualize the heat maps of absolute difference in output predictions when we occlude small portions in the image. The results are shown in Figure 3.3. These images are from the same webcam taken at different times of the day. The heatmap shows which area has the most impact in determining the output sun altitude angle. When the sun is present in the first image, the model gives importance to that portion of the image. On the other hand, it gives importance to the reflective rock surface in the second image.

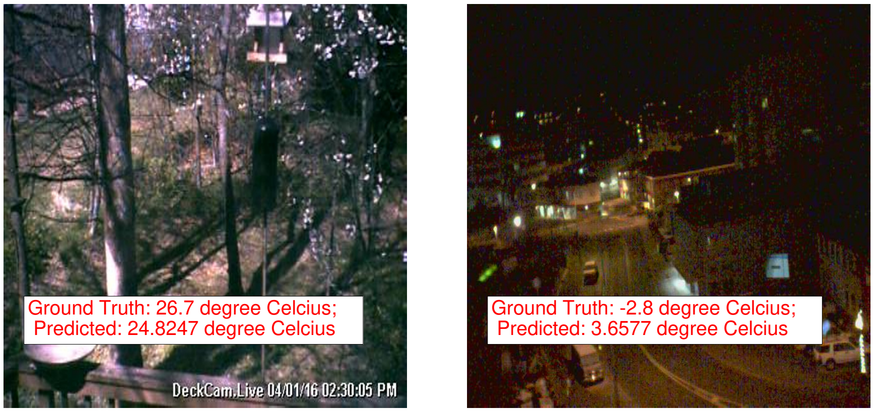

Ground Truth: 26.7 degree Celcius;
Predicted: 24.8247 degree Celcius
DeckCam.Live 04/01/16 02:30:05 PM
Ground Truth: -2.8 degree Celcius;
Predicted: 3.6577 degree Celcius

Figure 3.5: Ground truth vs the predicted temperature values for different scenes. The temperature model can predict temperatures even at night, which is not possible by the sun altitude angle model.

3.5.2 Temperature regression

We perform temperature prediction using an AlexNet based regression model with a mean absolute loss layer. The average temperature error is 8.94C and the Pearson correlation between the ground truth and predicted temperatures is 0.7339. For the ResNet based model, the RMS error reduces to 7.45C for the L2 loss based model. Figure3.5 shows the ground truth and predicted temperature values from two different images.

Figure3.6 and 3.7 show that mean absolute loss performs better than mean squared loss. Figure3.6 shows that about 45% of images give less than 5C error and almost 80% give less than 10C error for mean absolute regression. Figure 3.7 plots the variation of average error with the actual ground truth label. The flatter or more uniform the curve, the better are the results. As we can see, mean absolute regression works better than mean squared regression. The Pearson correlation coefficient for

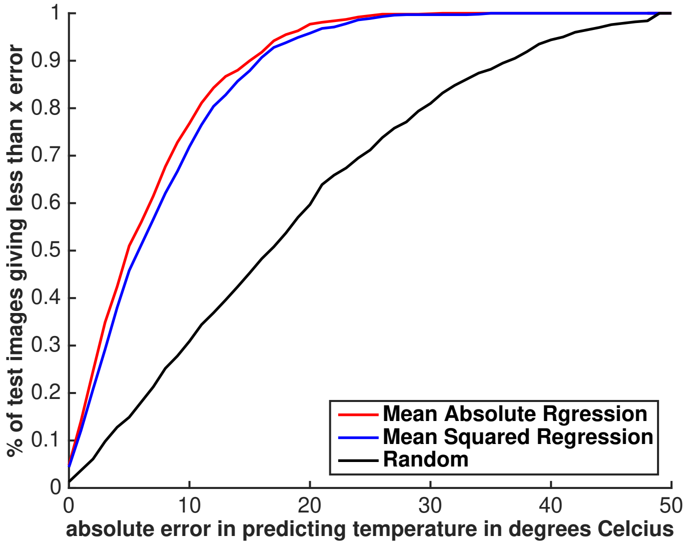


Figure 3.6: The x axis in the figure is the error in the prediction of temperature and the y axis is the percentage of test images giving error less than or equal to the corresponding x value. So about 80% of images have less than 10C error and about 45% gives less than 5C error for the mean absolute regression model. Also the mean absolute regression model performs better than the mean squared regression model.

mean absolute regression is 0.7339, whereas for mean squared regression is 0.6689. The RMSE for mean absolute regression is 8.94C whereas the RMSE for mean square regression is 9.83C.

3.5.3 Humidity regression

We find that although it is hard to infer the exact percentage of humidity from the image, there are usually some weather related visual cues that indicate the range of the humidity in the scene. Figure 3.8 shows examples of images that predict as low humidity (i.e. lass than 30 percent) and high humidity (i.e. greater than 85 percent). The numbers under the images are the regression results and the numbers in parentheses are the ground-truth humidity percentages. As shown in Figure 3.8, low humidity images in the first row are associated with clear skies; while in the

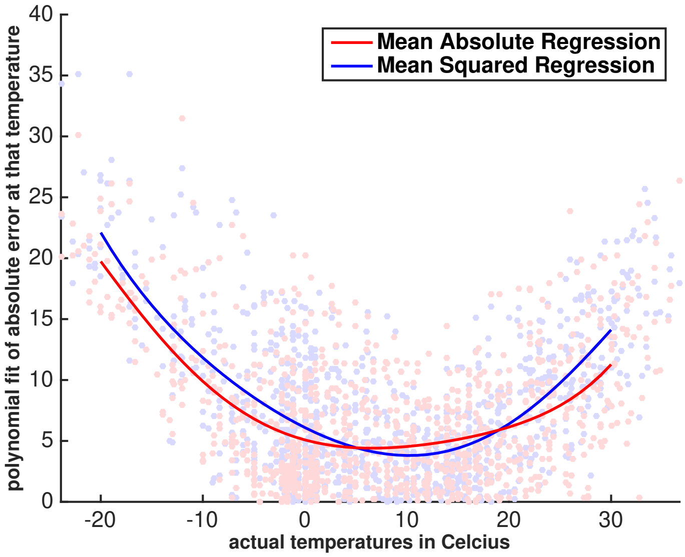
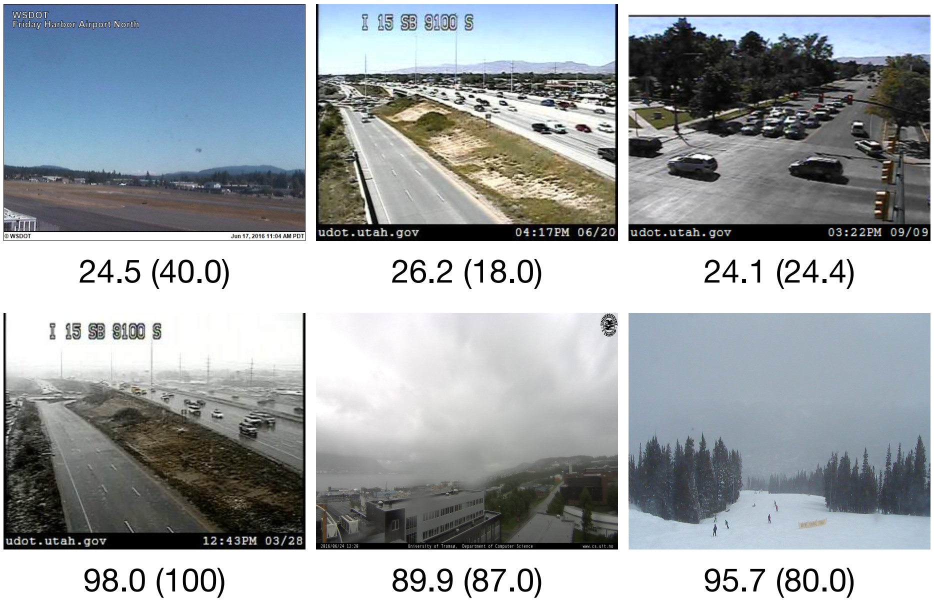


Figure 3.7: Polynomial fit to the error distribution vs. the ground truth temperature labels. The flatter the curve, the more uniform the error distribution across the output values. This shows that mean absolute regression performs better than the mean squared regression.

second row there can be rain, cloud, and snow indicating that the humidity values are high. Our regression network based on Alex Net achieves an average root mean square error (RMSE) of 18.42% whereas the Resnet based model achieves RMSE of 15.33%. Although the RMSE compared to the error in sun altitude angle and temperature regression is high, as shown in the following sections, our joint multi-task model can still benefit from the humidity information, which further improves the accuracy of metadata tampering detection.

3.5.4 Weather condition classification

Figure 3.9 shows the confusion matrix and some example classification results. The labels under the images are the output of the classifier and the labels in parentheses are the ground-truth labels. The red border indicates miss-classifications.



24.5 (40.0) 26.2 (18.0) 24.1 (24.4)

98.0 (100) 89.9 (87.0) 95.7 (80.0)

Figure 3.8: Example result of humidity regression. The number under each image is the predicted humidity and the number in the parentheses is the ground-truth. Top row: images predicted as having low humidity. Bottom row: images predicted as having high humidity. Although it is hard to predict the exact percentage of humidity from an image, there are usually some visual cues indicating the humidity range.

As shown in Figure 3.9, the classifier tends to classify rainy and snowy images as cloudy. This is because when it is raining or snowing, the sky looks cloudy as well. On the other hand, sometimes right after rain or snow, the road will look wet or covered with snow, which is why it is harder to separate these classes. Our classifier achieves 23.9% classification error rate on the test set after 100K training iteration.

3.5.5 Joint multi-task learning

Table 3.2 compares the classification error rate and regression RMSE on four different tasks with models learned separately and jointly with multi-task learning. All models are trained with the same network structure using the same hyper-

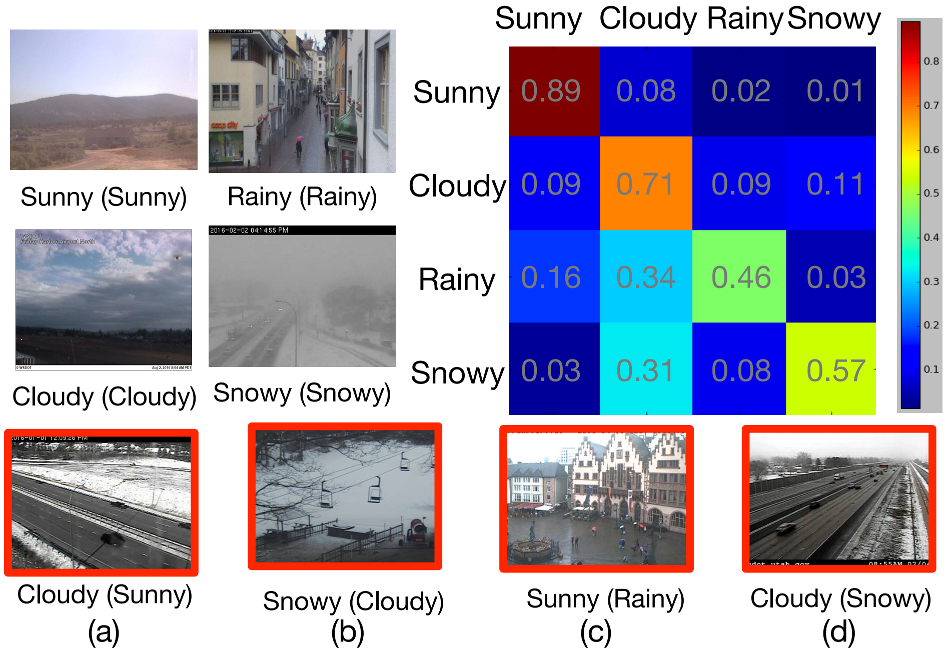


Figure 3.9: Example results and confusion matrix for weather condition classification. Red borders indicate misclassification. Rainy and snowy are prone to be misclassified as cloudy because the sky in each image is cloudy as well. (a) Sunny images misclassified as cloudy because the sky, which is an important cue for sunny images, is not visible. (b) Cloudy image misclassified as snowy because the snow covers a huge percentage of the image. (c) Rainy image misclassified as sunny because of the bright sky. (d) Snowy image misclassified as cloudy because the snow on the highway is mostly removed. Overall, our model can achieve 28.3% classification error rate.

parameters with 100K training steps. As shown in the Table, all four tasks benefit from a joint model, with weather classification enjoying the highest relative improvement. This is probably because weather conditions are highly related to all three other tasks. After joint multi-task learning, we can achieve an RMSE of 10.81, 6.9, 15.09 for sun altitude angle, temperature, and humidity regression and an error rate of 23.9 for weather condition classification. In order to further analyze the benefit of multi-task learning for weather classification, we train three other models using one of the meteorological information sources (sun altitude angle, temperature, or

Task	Single	Joint	Rel. Improv.
Sun Angle (RMSE)	11.31	**10.81**	4.4%
Temperature (RMSE)	7.45	**6.90**	7.4%
Humidity (RMSE)	15.33	**15.09**	1.6%
Weather (ERR)	28.30	**23.90**	15.5%

Table 3.2: The RMSE and classification error rate of the individual models and the joint model. Joint multi-task learning can improve the results for all four tasks and yields the most significant improvement for the weather classification because the weather is highly related to the other three sources of information.

Model	Error Rate
Weather	28.30
Weather, sun angle	27.30
Weather, temperature	27.80
Weather, humidity	27.50
All	**23.90**

Table 3.3: Weather classification error rate, combining meteorological information and sun altitude angle. Each slightly helps to reduce the classification error rate, and best performance is achieved by combining all the information, which demonstrates the effect of multi-task learning.

humidity) as well as the weather condition as input labels. The results are shown in Table 3.3. Each type of meteorological information can slightly help with the weather classification, and the best performance is achieved by utilizing all of the meteorological information, which demonstrates the benefit of multi-task learning in meteorological information prediction.

3.6 Experiments on metadata tampering detection

To analyze the effectiveness of meteorological information on tamper detection, we generate different tampered datasets by changing the timestamps or the GPS locations on the test images. We use ROC curves and Area under ROC curves

Month (AUC)	Angle	Humidity	Temp.	Weather
1	53.5%	62.5%	63.5%	**67.7%**
2	61.5%	71.4%	68.7%	**72.5%**
3	67.9%	72.3%	**80.7%**	71.1%
4	76.9%	74.9%	**84.5%**	74.1%
5	81.6%	72.2%	**84.6%**	73.7%
6	83.8%	73.6%	**85.1%**	71.6%

Table 3.4: AUC on time tampered data with large time tampering, in the order of multiple months. The weather model yields the best performance when the tampered time is one to two months from the ground truth because other information only changes slightly during a short period of time. The temperature model achieves the best performance when the tampered time is three to six months off from ground truth due to seasonal temperature changes. Sun altitude angle prediction yields better performance when the tampered time is further from ground truth because the sun position changes for the same time of the day throughout different seasons.

(AUC) as our performance metrics. In the rest of this section, we discuss the results of tampering detection on different types of tampered test sets.

3.6.1 Time metadata tampering detection

We construct the time tampered dataset by changing the timestamps on half of the test images to create positive samples (i.e. tampered) while the rest of the test images maintain their authentic timestamps and serve as negative samples. The three types of time tampered datasets are constructed by changing the timestamps in the test images with different month, day, and hour variances respectively. We then use the absolute difference of the sun altitude angle, humidity, and temperature between the output of our model and the meteorological information downloaded from the Internet, as well as the weather probability score output to compute the ROC curve and the AUC percentage.

Day (AUC)	Angle	Humidity	Temp.	Weather
1	50.3%	60.1%	54.7%	**67.5%**
2	50.3%	61.6%	57.1%	**70.1%**
3	49.7%	64.8%	56.4%	**69.5%**
4	50.2%	63.9%	56.0%	**69.6%**
5	49.6%	64.7%	57.9%	**69.2%**
6	50.2%	64.8%	59.3%	**69.9%**

Table 3.5: AUC on time tampered data with time tampering in the order of multiple days. The weather model achieves the best performance compared to other models. This is because all other meteorological information and sun altitude angle only has little change during short periods and it is hard to detect the difference.

Table 3.4 shows the AUC using different types of meteorological information, on the time tampered dataset with tampering variation in months. As shown in the table, when the time difference is one to two months, the weather model has the best performance in detecting inconsistency. This is because the change in sun altitude angle, humidity, and the temperature is small and our model has a hard time perceiving differences in these properties. On the other hand, the weather classifier can better separate different weather conditions happening in different months. When the time difference is three to six months, the temperature model has the best performance, because there is seasonal change and temperature exhibits large differences, which can be detected by our model. Sun altitude angle performs better when the tampering time is larger because the sun angle at the same time of the day will change more with greater variation in months. The ROC curves for the temperature model based monthly time tamper detector are shown in Figure 3.10(a).

Table 3.5 shows the AUC on time tampering dataset with tamper quantity variation in days ranging from one to six days. The overall performance is worse

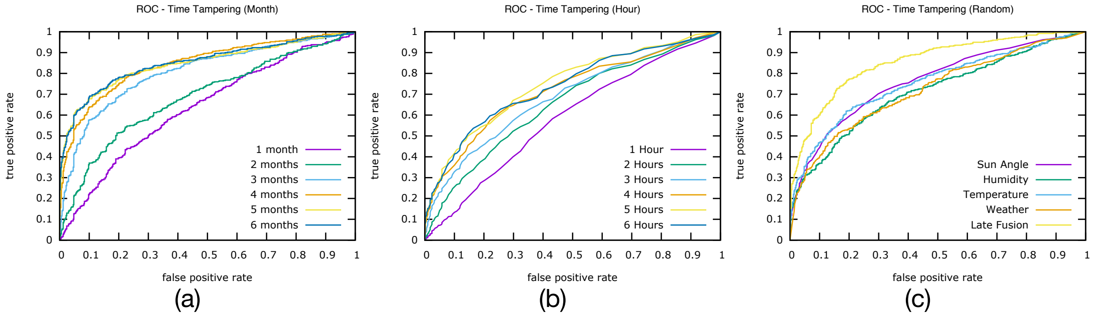


Figure 3.10: (a) ROC curves for the temperature model based monthly time tampered data detector. The different plots are for different variance noise in months added to tamper with metadata. The best performance expected is at a variation of 6 months, when the maximum seasonal variation is observed. (b) ROC curves for the sun altitude angle based hourly time tamper detector. The different plots are for different variance noise in hours that was used to modify the image time meta-data. The maximum sun altitude angle variation should be when the time difference is about 6 hours, which is what we can observe here. (c) ROC curve for time tampered data with timestamp changes ranging from one hour to one year. By combining all four models with late fusion, we can achieve better performance for tamper detection with an AUC of 85.5%.

than Table 3.4 because these meteorological measures exhibit less change during shorter intervals. The weather model again has the best performance overall and humidity has the second best.

Table 3.6 shows the AUC on the time tamper dataset with tampering in hours ranging from one to six hours. As shown in the table, sun altitude angle model has the best performance and the performance increases as the number of hours increase. This is because weather usually does not change too much during the same day, while sun angle will keep changing throughout the day. The ROC curve for the sun angle based hourly time tampering detection model is as shown in Figure 3.10 (b).

Since each of these models performs best when detecting different types of tampering, we combine them all with late fusion by adding the normalized scores

Hour (AUC)	Angle	Humidity	Temp.	Weather
1	**58.6%**	53.1%	51.2%	53.8%
2	**65.3%**	56.5%	53.5%	53.8%
3	**68.6%**	59.2%	54.3%	56.5%
4	**71.9%**	60.6%	54.5%	57.8%
5	**74.8%**	61.4%	56.4%	58.2%
6	**73.7%**	63.1%	56.6%	58.7%

Table 3.6: AUC on time tampered data with time tampering in the order of multiple hours. The sun altitude angle model yields the best performance because the sun altitude angle changes throughout the day while the weather usually varies little in a day.

from each model. Figure 3.10 (c) shows the ROC curve with a tampered dataset that randomly changes the timestamps within a range from one hour to one year. By combining all the models, we leverage the strength of each model and achieve better performance on tamper detection.

3.6.2 Location metadata tampering detection

We construct two location tampered test sets by changing the latitude and the longitude of the image metadata respectively. Figure 3.11 shows the AUC of tampering detection on longitude tampered test set with different models. The performance of each model increases as the tampered distance increases because of a larger change in meteorological features. However, the sun angle model does not perform well in this case, because 1000km is too short a distance to have any detectable sun angle variation.

Figure 3.12 shows the AUC of tampering detection on latitude tampered test set. Temperature model has better performance on this test set compared to the

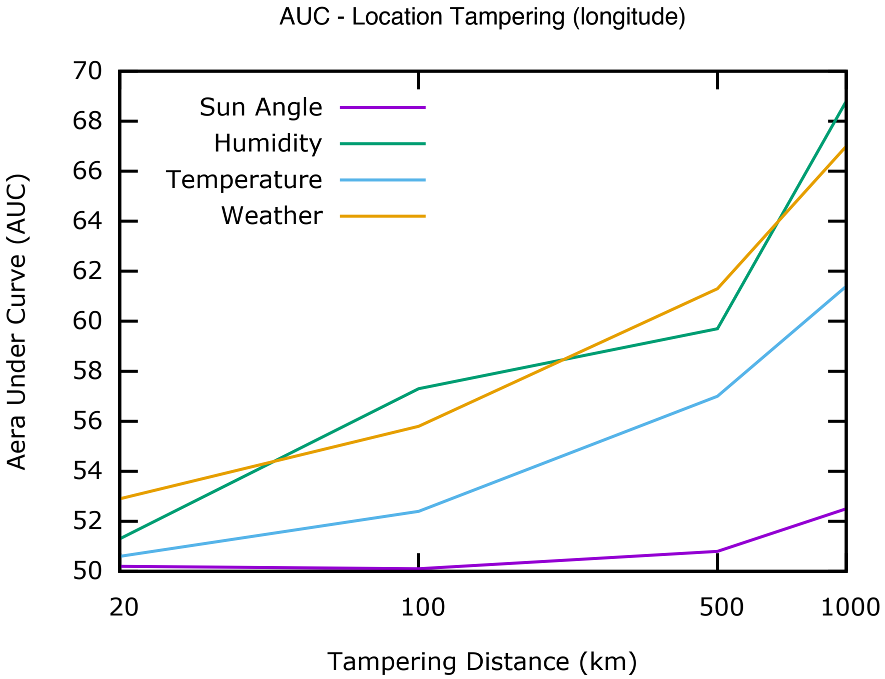


Figure 3.11: AUC on longitude tampered data. The performances of all models increase as the tampered distance increases.

previous one because temperature changes are more noticeable in different latitudes.

3.7 Conclusion

We propose a joint multi-task learning model to predict meteorological information from an image and use it to detect image metadata tampering. Our experiments show that joint multi-task model achieves better performance compared to any one model, and using the joint model we can detect different types of image metadata tampering with reasonable accuracy. Currently, we only apply simple late fusion to combine models for different meteorological information for tampering detection. Different ways to combine the models for meteorological information can be exploited in the future to further improve the detection results.

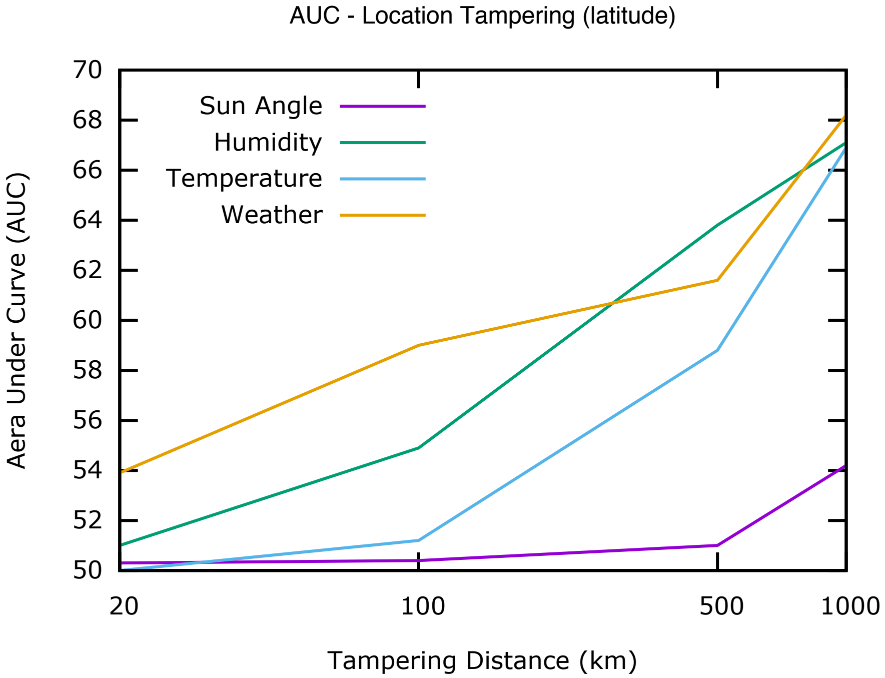
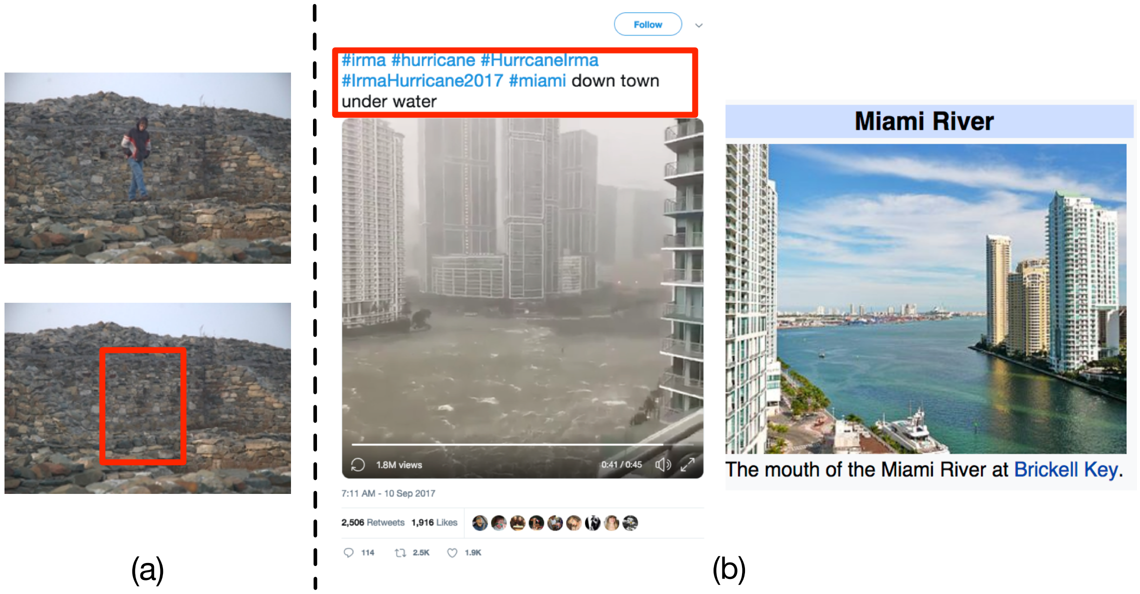


Figure 3.12: AUC on latitude tampered data. Compared to Figure 3.11, temperature model has better performance because temperature changes more drastically along different latitudes.

Chapter 4: Deep Representation Learning for Metadata Verification

4.1 Introduction

With the prevalence of social media platforms, an image shared on the Internet can reach millions of people in a short time. At the same time, people can easily share a tampered image for malicious goals such as creating panic or distorting public opinions with little effort. As a result, image tampering detection has become an emerging topic in the research community to prevent such attacks. Image tampering methods generally fall into two board categories: content manipulation and metadata tampering. The former alters the image content by splicing or removing some regions inside the image while the later doctors the metadata associated with the image, such as timestamp, geo-tag, or captions. Figure 4.1 (a) shows an example of content manipulation, where the person in the image is removed with image editing software; while Figure 4.1 (b) shows an example of metadata tampering, where the caption misleads people to believe a Miami downtown street is under water during hurricane Irma; but the water depicted in the video is, in fact, the Miami River. While many efforts have been made in the media forensics research community to develop algorithms to detect content manipulation with moderate success, little work has focused on detecting metadata tampering. We tackle the metadata verification


#irma #hurricane #HurrcaneIrma #IrmaHurricane2017 #miami down town under water

Miami River

The mouth of the Miami River at Brickell Key.

Figure 4.1: Two types of media tampering. (a) Example of content manipulation. The person in the image is removed with photo editing software. (b) Example of metadata tampering. The caption indicates a Miami downtown street is under water while the water depicts in the video is, in fact, the Miami River.

problem: given a set of images with common time and location metadata and some probe images, we want to verify whether the probe images have the same metadata. We focus on a specific type of metadata - event (i.e. time and places), which is the most common tampering target. Figure 6.1 illustrate the metadata verification problem.

Owing to the great success of deep convolutional neural networks in many related computer vision tasks such as image classification and image geo-localization, we attack metadata verification with a deep representation learning approach. We design a deep neural network called Attentive Bilinear Convolutional Neural Networks (AB-CNN) that aims to learn a relevant representation for the task.

We address several challenges in metadata verification with AB-CNN. First, we use an augmented training process to alleviate the problem of training data scarcity. To this end, we collect a large-scale image dataset with location information from Yahoo Flickr [41]. Second, a bilinear pooling layer is included to distinguish different fine-grained events. Finally, an attention module is utilized to help the model learns to focus on important visual cues and ignore diverse visual contents that are irrelevant to event verification.

The contributions of this work include: (1) To our best knowledge, this is the first work to address metadata verification with deep representation learning approach. We show how a deep neural network can be adapted for metadata verification, and the proposed AB-CNN is able to learn relevant representation for the task. (2) We construct a large-scale image dataset with 1 million images from 1,000 different location downloaded from Yahoo Flickr, which we show is useful for learning a representation for metadata verification. (3) In Section 4.4, we show that AB-CNN can be extended to the landmark recognition task, and achieving state-of-the-art performance in Google Landmark Recognition Challenge.

4.2 Related Works

Content manipulation detection traditionally focus on detecting tampering artifacts within the image such as double JPEG compression [42], CFA color array analysis [43], and local noise pattern analysis [44, 45]. Recently, with the advance of deep learning research, many works also adopt deep learning for content manipulation detection. Specifically, Cozzolino [46] combine the steganalysis rich

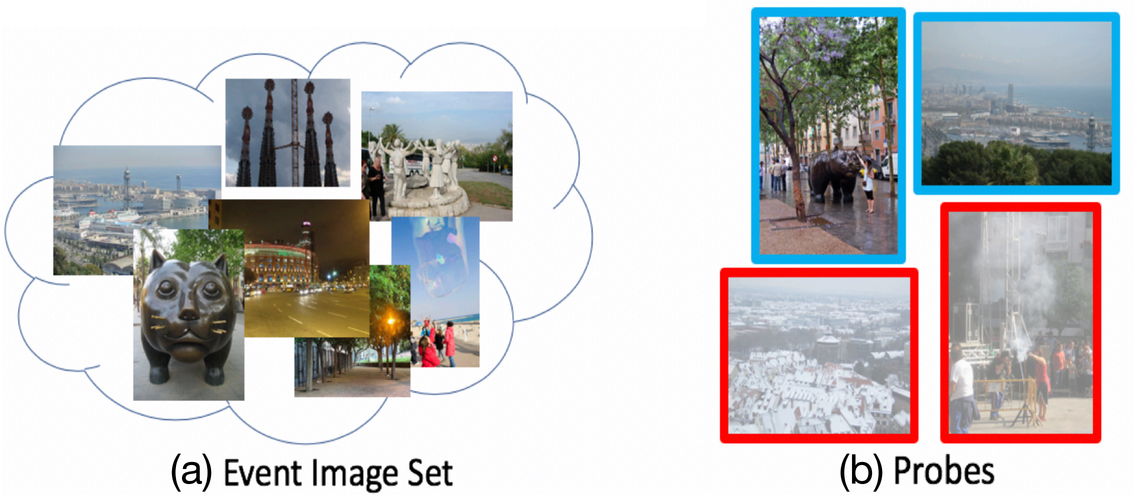


Figure 4.2: Given (a) a set of images with the same metadata (i.e. taken from a known event) and (b) some probe images, the goal of metadata verification is to verify whether these probe images have the same metadata (i.e. taken in the same event). In the above examples, images with the blue border are positive samples and images with the red border are negative samples.

model (SRM) with a convolutional neural network (CNN) for localizing manipulated regions. Rao [47] uses SRM filters as initialization for CNN kernels for manipulation detection. Salloum [48] use a fully convolutional network (FCN) to directly predict the tampered region. Bappy [49] use a sequence model with LSTM to find boundary artifacts. Zhou [50] adopt a two-stream network with faster-rcnn [51] framework for detecting manipulated regions. We also adopt a deep learning model for tamper detection; however, we focus on metadata tampering instead of content manipulation detection. Some works also utilize metadata [52, 53] to improve detection accuracy. However, they assume metadata is always trustworthy, which also shows the importance of metadata tampering detection.

Metadata tampering detection aims to verify the authenticity of the metadata associated with the image. Kakar [34] propose an algorithm for sun azimuth

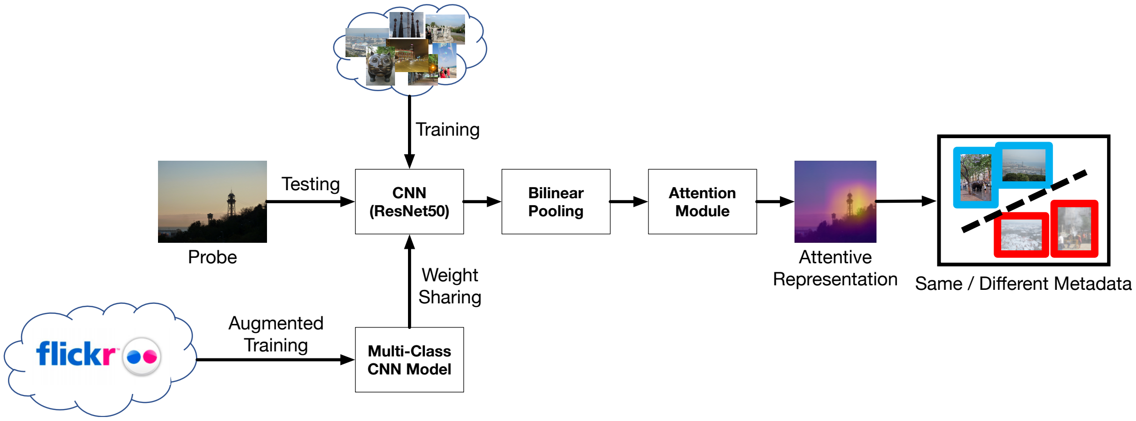


Figure 4.3: System overview of the proposed network configuration. The system contains three main modules. (1) An augmented training set is collected from the Internet and used to pre-train a multi-class for transfer learning. (2) Bilinear pooling is adapted to model the feature correlation. (3) An attention module is utilized to automatically learn to handle the diverse visual content within an event image set. Section 4.3 provide a detail description of the three modules.

estimation to verify temporal metadata. Li [54] use shadows to verify time and location. More recently, Chen [55] used a multi-task learning framework to predict various meteorological information in order to detect discrepancies between image content and metadata. In constrast to previous works, we use a deep representation learning approach to learn a suitable representation for metadata verification directly from data.

Landmark Recognition aims to recognize popular landmarks depict in images. Different approaches such as local feature matching [56], image retrieval [4, 57], and image classification [58] have been adopted for the task. We show that our model can also be extended to landmark recognition and achieving a competitive result in the Google Landmark Recognition Challenge.

4.3 Methods

4.3.1 System Overview

We cast the metadata verification as a representation learning problem and propose a convolutional neural network called Attentive Bilinear CNN (ABC-Net) with binary outputs as our model. Figure 6.4 shows an overview of the proposed network. It includes three modules that are designed for verifying a specific type of metadata – event. First, a multi-class image classification model is pre-trained with an augmented training set and shares weights with the binary model. Second, bilinear pooling is adopted to model feature correlations. Third, an attention module is learned for the model to focus on informative image regions. The remainder of the section describes the details of each module.

4.3.2 Augmented Training and Weight Sharing

Because it is hard to collect many images having nearly identical location and time , the metadata verification task typically has less training images compared to traditional image classification. To address this limitation, we utilize a transfer learning technique that is widely adopted in the literature – we fine-tune a pretrained network that is previously trained on a multi-class image dataset. Previous work usually pretrains the model on ImageNet [17], which contains around one million images from 1,000 different classes. However, we find that images from the ImageNet dataset are usually quite different from the images used in the metadata verification task. ImageNet contains images of different objects while metadata verification datasets contains images of different scenes. To this end, we collected a large-scale image dataset consist of one million images taken in 1,000 different locations as our augmented training set. The dataset is a subset of YFCC100M [41]. For each image in YFCC100M dataset with geo-tag, we first obtain a hierarchy of Where-On-Earth Identifier (WOEID) by reverse geocoding, and then select the finest scale of the WOEID as its location label. We first download 1,000 images from each of the 1,000 different locations containing the most images. We then run an indoor/outdoor classifier [59] on the downloaded images and remove indoor images from the dataset. We only keep locations with more than 500 outdoor images, so the final dataset consists of 914,109 images from 995 different locations. We call this the Yahoo Flickr Location (Yahoo-FL) Dataset. Figure 4.4 shows some example images in the Yahoo-FL dataset.

4.3.3 Bilinear Pooling

Bilinear pooling has shown promising results on several computer vision tasks, such as fined-grained classification, texture classification, and visual question answering [60, 61, 62, 63]. We hypothesize that it will also be helpful for the metadata verification since images from different events sometimes might contain similar but different fined grained visual patterns, and when an attacker tampers the metadata, they will usually choose a different but similar event. For instance, Figure 4.5 contains several examples of similar structures with fine-grained texture differences

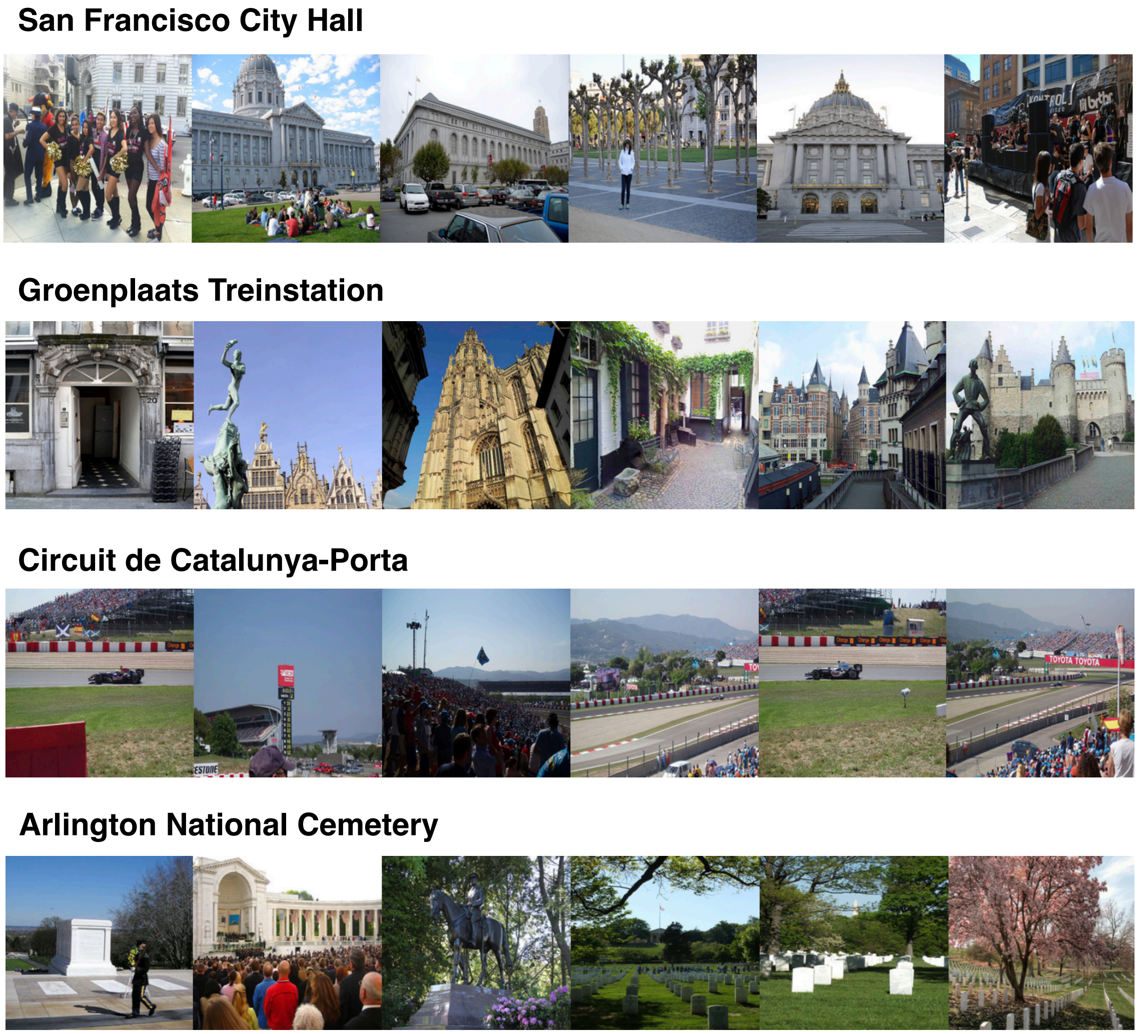


Figure 4.4: Example images in the Yahoo-FL dataset. The dataset consists of 955 different locations with 914,109 outdoor images.

from different locations.

Given an input image I, we first use a convolutional neural network (CNN) to extract a feature map $F \in R^{w \times h \times c}$. We then can calculate the bilinear features as:

$$B(F) = \sum_{i=1}^{W} \sum_{j=1}^{H} F_{ij} F_{ij}^{T} \tag{4.1}$$

As described in [62], this formulation is related to order-less descriptors such as

VLAD [64], Fisher vectors [65] and region covariance [66], and is able to effectively capture the second order statistics of the input features. The diagonal entries of the output bilinear matrix $B(F)$ represent the variances of each feature channel, and the off-diagonal entries represent the correlations between different feature channels. The output matrix $B(F)$ is first converting to a vector v and then goes through following mapping function:

$$f(v) = \frac{sign(v)\sqrt{|v|}}{||sign(v)\sqrt{|v|}||_2}, \tag{4.2}$$

which calculates the signed square root followed by l-2 normalization to project the vector to Euclidean space [65]. Note that for an input feature with c channels, the dimensionality of the above bilinear representation is c^2. Such high dimensionality prevents efficient training of the deep neural networks due to computation and memory constraint. Therefore, following previous work [60], we use tensor sketching to reduce the dimensionality of the original bilinear vector. We first generate random vectors $h_1, h_2 \in N^c$ and $s_1, s_2 \in \{-1, +1\}^c$, where $h_k(i)$ is uniformly drawn from $\{1, 2, ..., d\}$, $s_k(i)$ is uniformly drawn from $\{-1, +1\}$, and d is the target dimension. We then define the sketch function $\Psi(x, s, h) = [S_1, S_2, ..., S_d]$ as:

$$S_j = \sum_{i:h(i)=j} s(i)x_i, \tag{4.3}$$

Figure 4.5: Similar structures from different locations in the Google landmark recognition dataset. Bilinear pooling describes in Section 4.3.3 helps to distinct the fine-grained differences between them.

finally, we compute the compact bilinear vector as:

$$\phi(x) = F^{-1}(F(\Psi(x, s_1, h_1)) \circ F(\Psi(x, s_2, h_2))), \tag{4.4}$$

where F and F^{-1} represent the Fast Fourier Transformation and Inverse Fast Fourier Transformation, and $\circ$ represents element-wise multiplication. For a CNN feature map F, the compact bilinear feature representation is calculated as:

$$\Phi(F) = \sum_{i,j} \phi(F_{ij}). \tag{4.5}$$

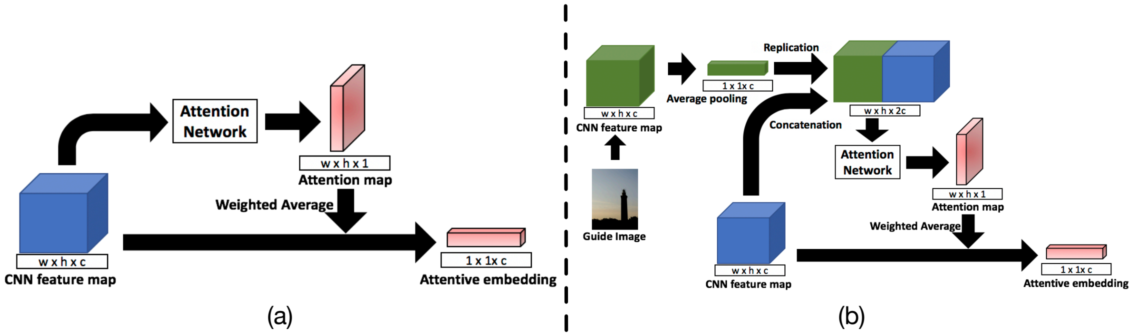


Figure 4.6: (a) Self-attention module. Given a convolutional feature map, the attention network uses it to compute a spatial attention map. The attention map is then used as weights for pooling the original feature map. (b) Guided-attention module. The model takes another image from the same event as the guide image, and uses it to provide additional information for learning the attention function.

4.3.4 Attentive Bilinear CNN (AB-CNN)

For metadata verification, images from the same event usually contain diverse visual contents beside information which can be used to identify the event; additionally, the useful information usually located in a small region of the image. To this end, we propose to utilize an attention module to the network to better focus on the informative regions in the image. Attention models have been successfully applied to many different tasks such as machine translation [67], image captioning [68], visual question answering [69], and image retrieval [57]. We first describe a self-attention module commonly used in previous works, and we then describe a novel attention module, guided-attention, integrated with our bilinear CNN model for the metadata verification task. Figure 4.6 illustrates the proposed attention module.

4.3.4.1 Self-Attention

Given a CNN feature map $F \in R^{w \times h \times c}$, the goal of the attention module is to learn a attention function $\alpha(x; \theta)$, where $\alpha(F_{ij}; \theta)$ represent the importance of the region (i, j) in the image, and θ denotes the parameters of the attention function. The final representation is the weighted sum of the feature map instead of unweighted average pooling:

$$\Phi(F) = \sum_{i,j} \alpha(F_{ij}; \theta) F_{ij}. \tag{4.6}$$

Here the attention function α is represented by an attention network with two 1×1 convolutional layers. To avoid negative weighting, we apply a soft-plus function to the attention output similar to [57]. Finally, by combining with compact bilinear pooling, the final attentive bilinear representation is calculated as:

$$\Phi(F) = \sum_{i,j} log(1 + e^{\alpha(F_{ij}; \theta)}) \phi(F_{ij}). \tag{4.7}$$

4.3.4.2 Guided-Attention

As shown in Figure 4.7, sometimes an image might contain multiple informative regions, and which region in the image is important for identifying the event depends on other images in the dataset. We propose a novel attention module called guided-attention to address this issue. As illustrated in Figure 4.6 (b), during training, the model takes an extra guide image from the same class as input, and calculates its convolutional feature map with average pooling g. The attentive bilinear representation is then computed as:

$$\Phi(F) = \sum_{i,j} log(1 + e^{\alpha(F_{ij}, g; \theta)})\phi(F_{ij}), \tag{4.8}$$

for each spatial location, g and F_{ij} are concatenated and go through the attention network to compute the attention score. Note that during test time, there is no guide image available, since we do not know the event of the probe image; therefore, we use the probe image itself as the guide image. We empirically observe that works well, as it provides additional global context to the attention function.

4.3.4.3 Attention Learning

The attention function is learned jointly with the CNN network parameters. For each input image, the network will calculate a vector $y \in R^2$:

$$y = W\Phi(F), \tag{4.9}$$

where $W \in R^{2\times d}$ are the network parameters of the final classification layer. The soft-max cross entropy is used to calculate the loss for training:

$$L = -y^* log\Big(\frac{e^y}{1^T e^y}\Big), \tag{4.10}$$

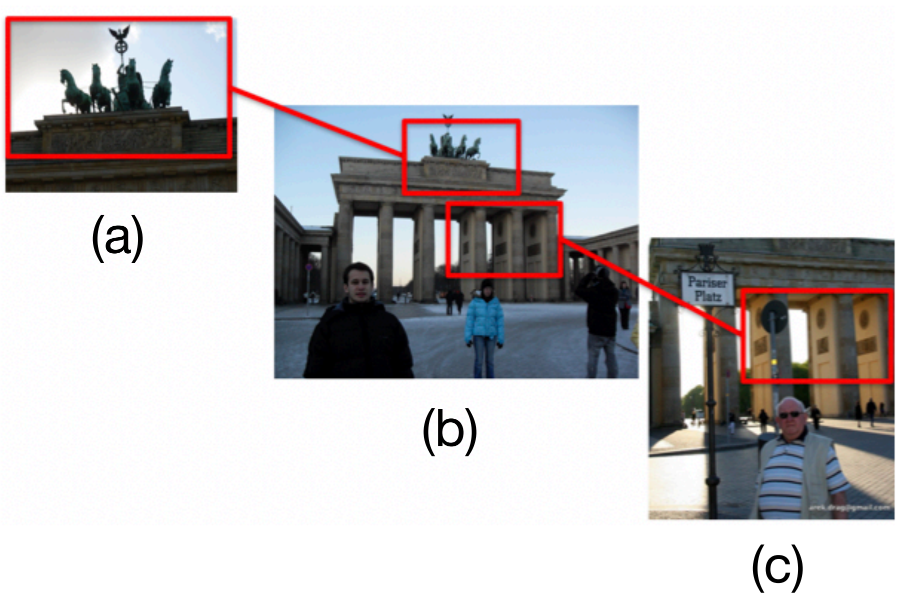


Figure 4.7: Some images have multiple informative regions. For instance, (b) contains two regions that can help to identify the event. When comparing with image (a), the upper part is more important; when comparing with image (c), the lower part is more important for learning good representation.

where y^* is the ground-truth label. The network is trained with stochastic gradient descent with back-propagation to minimize the above loss function.

4.4 Experiments

4.4.1 Datasets and Evaluation Metrics

We evaluate the proposed method using three different datasets, including (1) Yahoo Flickr Location Dataset (**Yahoo-FL**), (2) Medifor Event Verification dataset (**MediFor-EV**), (3) Google Landmark Recognition Dataset (**Google-LR**).

Yahoo-FL contains 914K images from 955 locations, and it is a subset of YFCC100M dataset [41]. Details of the dataset can be found in Section 4.3.2. Figure 4.4 shows some images in the dataset. Each location represents one event in

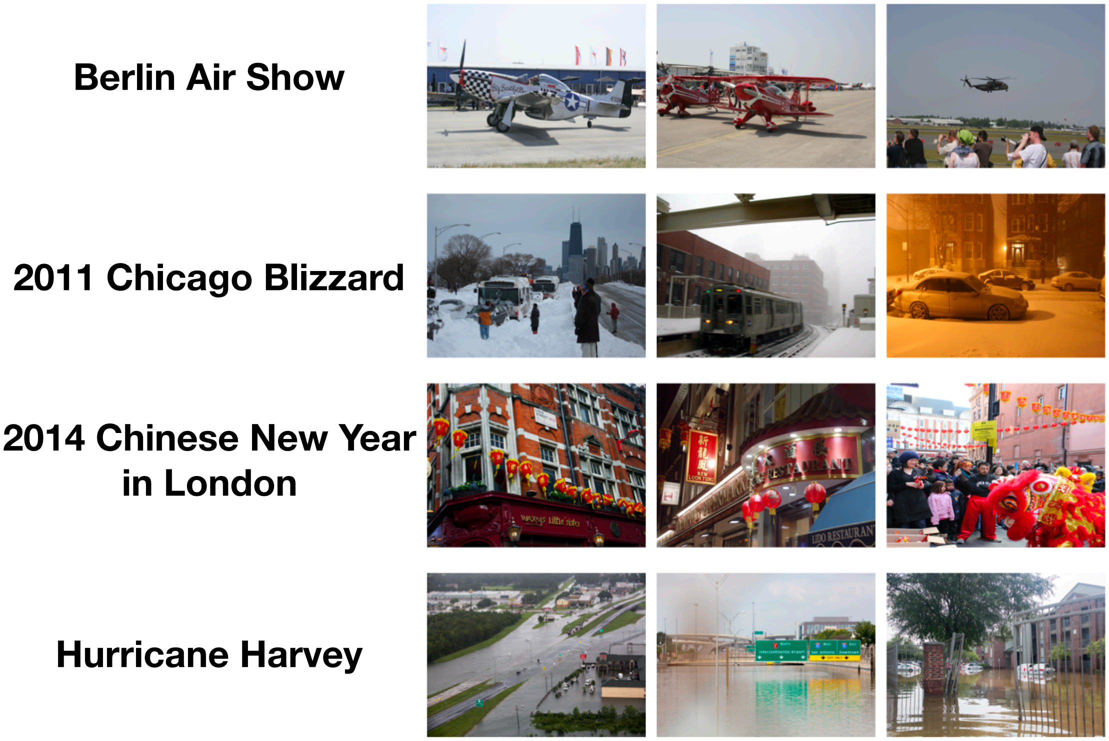


Figure 4.8: Images in the MediFor-EV dataset. The dataset contains 2,315 images from twelve different events. Each column contains images of the same event.

our experiments, and we select 35 locations as our event set while other locations are used as augmented training set. For each event set, we randomly select 50 images for testing and up to 1000 images for training. We also randomly select an equal number of images as negative samples for both training and testing. Note that negative samples in the test set are selected from a disjoint set of locations to test the generalization ability of the model. For Yahoo-FL, we use binary classification accuracy (ACC) as our evaluation metrics.

MediFor-EV consists of images taken at twelve different events throughout the world, including Berlin Air Show, Berlin Marathon, 2011 Chicago Blizzard, 2014 Chinese New Year at London, Hurricane Harvey, Hurricane Ike, Hurricane Irma, Hurricane Katrina, Hurricane Matthew, Hurricane Sandy, 2010 Oshkosh, and

2011 Oshkosh. Figure 4.8 shows some example images in the dataset. The dataset contains two splits, training set, and testing set. The training set contains around 200 images of each event with a total of 2,315 images, and the test set contains 50 images of each event with a total of 600 images. The dataset is part of the Media Forensics Challenge 2018[1]. For MediFor-EV dataset, we follow the challenge guideline and use area under ROC curve (AUC) as our evaluation metric.

Google-LR contains 1.2 million training images from 15K different landmarks and 117.7K probe images. The dataset is extremely imbalanced with 6.5K classes in the training set contain fewer than 10 images. The dataset is released as part of the Google Landmark Recognition Challenge[2]. We follow the competition guideline and use global average precision (GAP) as our evaluation metric. Global average precision considers the prediction as well as the confidence score of all probe images, and is calculated as:

$$GAP = \sum_{i=1}^{N} p(i) r(i), \tag{4.11}$$

where N is the number of probe images, $p(i)$ is the precision at rank i, and $r(i)$ denotes the relevance of prediction i. Detailed description of this evaluation metric is available on the Google Landmark Recognition Challenge website [3].

[1] https://www.nist.gov/itl/iad/mig/media-forensics-challenge-2018

[2] https://landmarkscvprw18.github.io/

[3] https://www.kaggle.com/c/landmark-recognition-challengeevaluation

Method	Accuracy
CNN (ImageNet)	69.0%
CNN (Flickr)	83.4%
A-CNN	83.7%
B-CNN	83.7%
AB-CNN	**84.5%**

Table 4.1: Experimental results of Yahoo-FL dataset. Augmented training provides large performance gain as the model learns better feature for distinguishing different events. Both A-CNN and B-CNN provide moderate improvement. When combining both, the model achieves the highest performance.

4.4.2 Compared Algorithm

We compare the proposed methods with following the baseline algorithms: (1) **CNN(ImageNet)**: Basic CNN with ResNet-50 [37] architecture pre-trained on ImageNet Dataset. (2) **CNN(Flickr)**: ResNet-50 with augmented training on Yahoo-FL as described in Section 4.3.2. (3) **A-CNN**: ResNet-50 with augmented training and attention module described in Section 4.3.4. (4) **B-CNN**: ResNet-50 with augmented training and bilinear pooling described in Section 4.3.3. (5) **AB-CNN** ResNet-50 with augmented training, bilinear pooling, and attention module. Section 4.3.4.1.

4.4.3 Experimental Results on Yahoo-FL

Table 4.1 shows results with different algorithms on the Yahoo-FL dataset. By using the augmented training set, the accuracy greatly improves from 69.0% to 83.4%. This is because the model learns a representations which better separates different events through the multi-class classification training process. By adding either attention module (A-CNN) or bilinear pooling layer (B-CNN), the

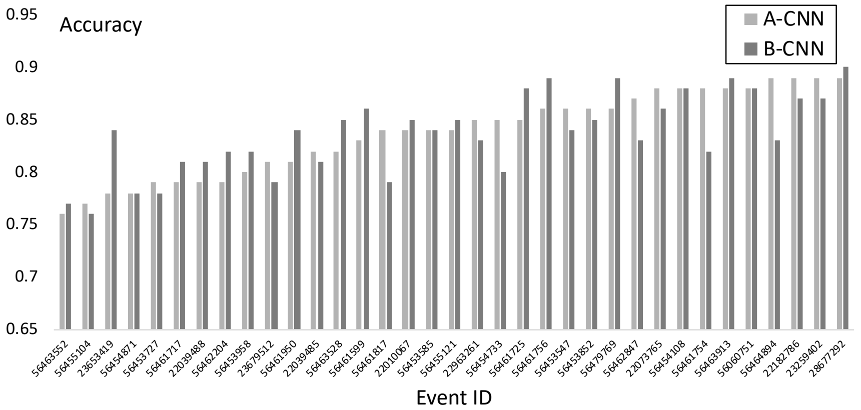


Figure 4.9: Binary classification accuracy of Yahoo-FL for different events using A-CNN and B-CNN. Each model improves performance on different events. Two model are complementary to each other. AB-CNN combine the two model together, and it is able to benefit from both modules, which further improve the overall accuracy.

performance improves from 83.4% to 83.7%. Figure 4.9 shows classification accuracy broken down to different events for A-CNN and B-CNN. A-CNN and B-CNN improve performance on different events, and by combining the two methods together, the performance further increases to 84.5%. The results demonstrate the effectiveness of the proposed network and the complementary effect of bilinear pooling and attention module. Figure 4.10 shows qualitative results with the attention heatmap overlayed on the test images. The attention module helps the model focus on background structure which is helpful to distinguish different events.

4.4.4 Experimental Results on MediFor-EV

Table 4.2 shows the experimental results on MediFor-EV. Similar to the results on Yahoo-FL, the proposed method is able to learn better representations and achieve higher performance compared to the baseline CNN model. Augmented training provides less performance gain for MediFor-EV dataset compared to Yahoo-

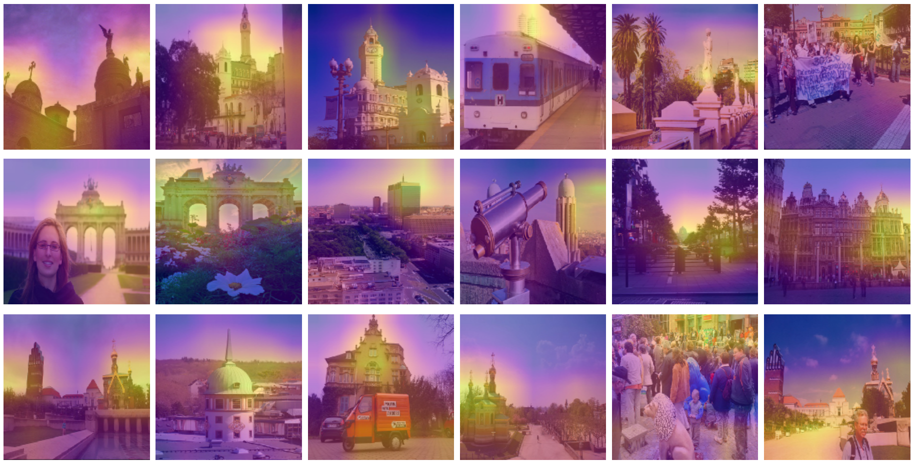


Figure 4.10: Visualization of spatial attention on the test image. Our model learns to focus on the background structure instead of people in the foreground, which helps distinct different events.

Method	AUC
CNN (ImageNet)	84.4%
CNN (Flickr)	87.2%
A-CNN	87.8%
B-CNN	87.5%
AB-CNN	**88.9%**

Table 4.2: Experimental Results in MediFor-EV dataset. AB-CNN improves the performance from 84.4% to 88.9% in terms of area under ROC curve (AUC) compare to baseline CNN.

FL, because the data distributions between the augmented training set and event training set have large differences. Note that since the amount of training data in MediFor-EV is small, we fix all weights in CNN except the final classification layer after the augmented training process. Figure 4.11 shows some positive results on MediFor-EV with the proposed model. Our model can learn an appropriate representation and successfully verify the metadata in these examples. Figure 4.12 shows some failure cases of our model. Probes with close-up images and indoor images are exceedingly challenging because there is usually not enough visual information for the model to learn a good representation.

Method	GAP
CNN (ImageNet)	14.0%
A-CNN	14.8%
B-CNN	15.4%
AB-CNN	**18.7%**
AB-CNN (SE-ResNeXt) + Re-Ranking	25.6%
AB-CNN (Ensembles) + Re-Ranking	**32.2%**

Table 4.3: Results of AB-CNN on Google-LR dataset. We extend the AB-CNN for landmark recognition task and improve the result from 14.0% to 18.7% in terms of global average precision (GAP) compare to the baseline model. By combining advanced network architecture, spatial local feature re-ranking, and model ensemble, we improve the GAP to 32.2%, which give us rank 2 out of 483 teams in the challenge.

4.4.5 Experimental Results on Google-LR

We extend AB-CNN for the landmark recognition task by considering the landmark id of each image as its metadata. Since Google-LR already consists of one million training images, we do not apply the augmented training process with Yahoo-FL. Instead, we directly train a multi-class classification network with the training data. Table 4.3 shows the experimental results on Google-LR. By adapting AB-CNN to the landmark recognition task, we were able to improve the GAP from 14.0% to 18.7% compared to the baseline CNN approach. By adopting a more advanced network architecture, SE-ResNeXt [70], with spatial re-ranking using local features [57], we improved the GAP to 25.6%. Finally, by combining multiple models with ensembling, we can further improve the GAP on the validation set to 32.2%.

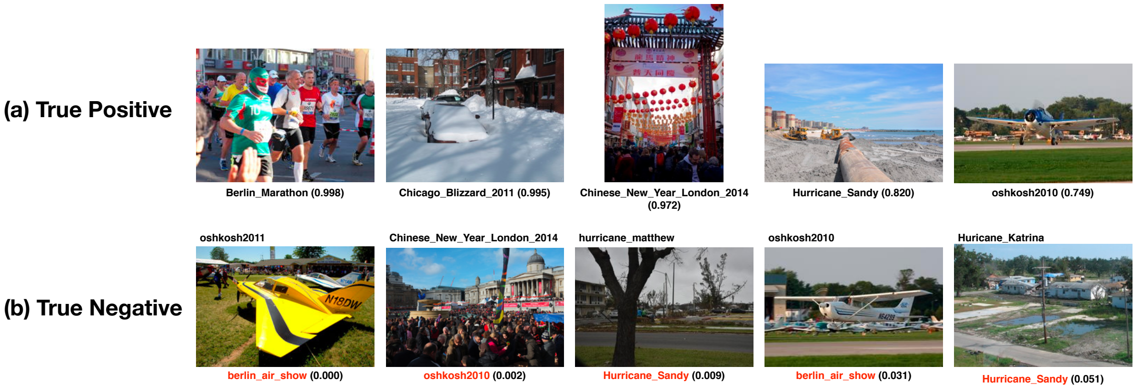


Figure 4.11: Positive results generated by our model. The event name under the image is the metadata associated with the images. Name in red indicates the metadata was tampered and the event name on the top is the original metadata. Numbers in the parenthesis are the confidence scores output by our model. (a) True positive probes verified by our model. Our model can extract informative representation from the image and use it to verify the metadata. (b) True negative examples. Our model successfully rejects these tampered probes based on the learned representation.

4.5 Conclusion

We address the metadata verification with a deep representation learning approach. Based on experiments on three different datasets, we show the proposed network configuration, AB-CNN, can learn suitable representation for the task and verifying a common metadata, event, associated with the image. We also show our model can be extended to landmark recognition, achieving state-of-the-art performance. Metadata verification is an important topic for media forensics researches, future directions includes metadata verification with a wider range of metadata, as well as detecting inconsistencies between different metadata in addition to media content.

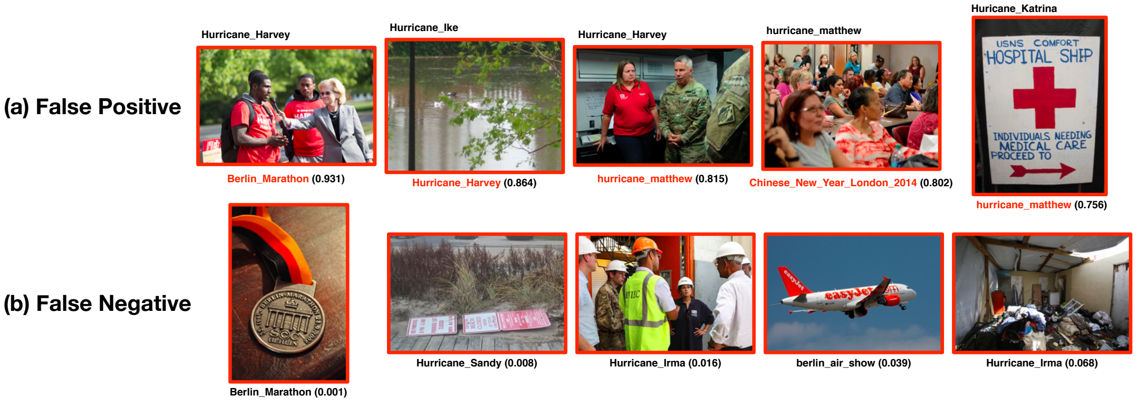


Figure 4.12: Negative results generated by our model. (a) False positive examples. Our model fails to detect this tampered probe. (b) False negative examples. Our model indicates the metadata was tampered while it is actually not. Close-up image (Last example in the first row, first and second example in the second row) and indoor image (third and fourth example in first row, third and fifth example in the second row) is especially challenging because these probe images usually does not contain enough information for the model to verify the metadata.

Chapter 5: Toward Realistic Image Compositing with Adversarial Learning

5.1 Introduction

Image compositing aims to create a realistic-looking image by taking the foreground object of one image and combining it with the background from another image (cf. Figure 6.1). In order to make the composite image look realistic, many factors need to be considered, such as scene geometry, object appearance, and semantic layout. It is a challenging task and usually requires a human expert carefully adjusting details including geometry and color using professional image editing software such as PhotoShop to create a single composition.

Many previous works [71, 72, 73, 74, 75, 76, 77, 78] try to alleviate the burden by creating algorithms that can automatically adjust the appearance of the foreground image and makes it fit into the background naturally. While this may work in some cases, many of these approaches still require human supervision to help with tasks such as determining the location and size of the foreground object or capturing the lighting conditions of the scene.

Generative adversarial networks (GANs) have recently been shown to have the ability to generate realistic looking images [79, 80, 81, 82, 83, 84, 85, 86] by learning to deceive an adversarially trained discriminator network. However, image composition is a different task from image generation because the composite image must maintain details from the input images and apply only slight changes to improve the realism of the composition. Recent work [87] modified the GAN framework by restricting the range of the generator to a geometric manifold using a spatial transformer network [88], in order to generate realistic composite images that are geometrically consistent. However, such a model only works if the foreground appearance is already consistent with the background image. If the domain of the foreground and the background images are different, geometric transformation alone does not have the ability to generate a natural-looking composite image. As shown in Figure 6.1, for a composite image to be realistic, the model needs to address both geometric and color consistency. However, it is not trivial to combine previous works to automatically adjust both color and geometry since these two properties are interdependent: geometric correction relies on color consistency while color correction also relies on geometric consistency.

To address the above issue, we propose a GANs architecture called Geometrically and Color Consistent GANs (GCC-GANs) for image compositing that simultaneously learns both geometric and color correction with adversarial training. GCC-GANs contain four sub-networks, a transformation network, a refinement network, a discriminator network, and a segmentation network. The transformation network and the refinement network act together as the generative compositing model, which aims to generate a composite image while considering geometric, color, and boundary consistency. At the same time, the discriminator network and the segmentation network help to increase the realism of the composite image through adversarial training. The discriminator network learns to separate the composite images from the real ones while the segmentation network learns to separate the foreground object from the background in the composite images. Unlike previous works that restrict the generator to geometric transformations, our model can apply both geometric and color correction as well as boundary refinement to generate a composite image. GCC-GANs are trained end-to-end with a geometric loss, a appearance loss, a adversarial loss, and a adversarial segmentation loss. Experimental results show that it can generate geometrically and color consistent images in both synthetic and real-world datasets.

The contributions of this paper include: (1) Demonstrating the need for both geometric and color consistency for the image compositing task, (2) proposing an novel end-to-end model that creates realistic composite images based on the generative adversarial network framework, and (3)extensive evaluations including human perception experiments show the ability of the proposed model to generate realistic composite images compare to different state-of-the-art methods.

5.2 Related Work

Image Compositing models try to combine a foreground image with a background image seamlessly. Many prior works focus on how to modify the appearance of the foreground image to better fit into the background based on color gradients

[71, 72] or color statistics [73, 74, 75]. Agarwala [89] provide a system to combine multiple source images taken in a similar scene. Lalonde [76] develop an interactive system to allow creating composite images by selecting foreground objects from a large database. With advancement of the deep learning research in computer vision, various deep learning models [77, 78, 87, 90] were also introduced for image compositing. Similar to our approach, Zhu use a discriminative model to estimate the realism of composite images. However, their discriminative model is fixed during the image compositing process and can not be improved for better composition. Tsai introduced an end-to-end encoder-decoder network for image harmonization. Although these methods can generate realistic composition, they still rely on a human to complete the semantic tasks such as deciding the location and size of the foreground objects. Most recently, Tan [90] propose to use deep neural networks to learn the location and size of the foreground object for human composition; Lin [87] use generative adversarial networks (GANs) with spatial transformer networks [88] to learn the correct geometry transformation of the foreground object. These works consider geometric consistency in image compositing, but can only work when the domain of foreground image and background image are similar. Our work extends previous works by providing a unified end-to-end framework that learns to adjust both the geometry and appearance consistently, which allow our model to automatically compositing images from different sources.

There are also many works that try to combine synthetic 3d objects with images [91, 92, 93, 94, 95]. However, these methods require explicitly reconstructing the scene geometry and environment illumination in order to render the 3d object.

On the other hand, our model can directly take the rendered object as input for composition.

Generative adversarial networks [79] have been utilized on many different image generation task [80, 81, 82, 84, 96, 97, 98, 99]. Conditional GANs [96] provide a way to generate images from different classes given different input. Isola [80] provide a framework that translate image from one domain to another given pairs of training images. Zhu [84] further extend the framework to work on unpaired training images using cycle consistency. However, these frameworks can not directly apply to image composition task since the composed images need to keep most of the the detail information of both foreground and background images in a consistent manner. Instead of direct image generation, our model utilize the adversarial training process to learn geometric and color corrections for realistic composition.

5.3 Proposed Method

5.3.1 System Overview

Figure 6.4 shows an overview of the proposed network architecture. The model consists of four sub-networks: a transformation network, a refinement network, a discriminator network, and a segmentation network. The transformation network and the refinement network act together as the generative compositing model and is described in section 5.3.2. The discriminator network and the segmentation network help the generative model thorough adversarial training and is described in section 5.3.3. Given an input triplet consists of a background image, a foreground image, and an object mask, the compositing model learns to composite realistic images; the discriminator network learns to distinguish composite images from the real images. At the same time, the segmentation network tries to separate the foreground object from the background in the composite image. The model is trained to optimize the min-max objective function described in section 5.3.4.

5.3.2 Generative Compositing Model

Given a foreground image with N pixels $I_f \in [0,1]^{N\times 3}$ with a foreground mask $\alpha \in \{0,1\}^N$ and a background image $I_b \in [0,1]^{N\times 3}$ as inputs $I = \{I_f, I_b, \alpha\}$, the process of image compositing can be formulated as follow: $\mathrm{I}_c = G(I;\theta_G)$

$= A(I) \circ F(I) + (1 - A(I)) \circ I_b$, where G is the compositing model which combines the foreground region of I_f indicated by the mask M and the background image I_b; θ_G is the model parameters. $F(I) \in [0,1]^{N\times 3}$ is the transformed foreground and $A(I) \in [0,1]^N$ is the alpha mask. Under this formulation, a simple alpha composition model can then be described as identity functions: $A(I) = \alpha; F(I) = I_f$.

If only the geometric correction is considered as in [87], the model becomes:

$\mathrm{A(I)} = \mathrm{H}(\alpha, T_h(I;\theta_G))$

$F(I) = H(I_f, T_h(I;\theta_G))$, where $H(\cdot)$ is the geometric transformation function, such as homography, affine or similarity transform, and $T_h(\cdot)$ the transformation matrix. We use spatial transformer network [88] to predict the transformation parameters.

On the other hand, if we assume foreground/background geometry is consistent and only consider the color correction, $F(I)$ becomes a color transformation function

$F(I) = C(I_f, T_c(I;\theta_G))$ which adjusts the appearance of the foreground image. we use a linear brightness and contrast model [77]:

$$C(I_f, T_c(I;\theta_G)) = I_f \mathbf{1} \lambda_1 000 \lambda_2 000 \lambda_3 \beta_1 \beta_2 \beta_3, \tag{5.1}$$

where $T_c(I;\theta_G) = (\lambda_1, \lambda_2, \lambda_3, \beta_1, \beta_2, \beta_3)$ is a transformation network that predicts the contrast and brightness parameters.

To apply both geometric and color correction to the composite image, we can then combine Equation 5.3.2 and Equation 5.1:

$$F(I) = C(H(I_f, T_h(I;\theta_G)), T_c(I;\theta_G)), \tag{5.2}$$

note that we can use a single network to predict both color and geometric transformation parameters at the same time, so that $T(I;\theta_G) = [T_h(I;\theta_G); T_c(I;\theta_G)]$ and simplify Equation 5.2 as:

$$F(I) = (C \circ H)(I_f, T(I;\theta_G)). \tag{5.3}$$

Equation 5.3.2 and Equation 5.3 together describe our compositing model $I_c = G(I;\theta_G)$. However, the composite image might still contain some boundary artifacts. To address this issue, we introduce a refinement network R with an encoder-decoder architecture that further refines the composite image. So the final composition model can be described as: $\mathrm{I}_c = G(I;\theta_G)$

$= R(A(I) \circ F(I) + (1 - A(I)) \circ I_b)$. We adopted architecture described in [100] for our refinement network.

5.3.3 Adversarial Training

Equation 5.3.2 describes our compositiing model $I_c = G(I; \theta_G)$ with transformation network and refinement network. We adopt a similar procedure described in [79] to train a discriminator network $D(x; \theta_D)$ with adversarial learning. Adversarial learning maximizes the following adversarial loss $\mathcal{L}_a$ to distinguish natural image I_b from the composite image I_c:

$$\mathcal{L}_a(D, G) = E_{I_b}\Big[logD(I_b)\Big] + E_{I_c}\Big[log(1 - D(I_c))\Big]. \tag{5.4}$$

We use a basic three-layer convolutional network for discriminator network and adopt spectral normalization [101] to stabilize the training process. To reduce the discrepancy between foreground and background in the composite image, we propose to train an additional segmentation network S that learns to separate the foreground object from the background in the composite image. The network is trained with adversarial segmentation loss: $\mathrm{L}_s(S, G) = \sum_{s \in fg} E_{I_c}\Big[log(1 - D_s(I_c))\Big]$ $+ \sum_{s \in bg} E_{I_c}\Big[log(D_s(I_c)\Big]$, where $s \in \{fg \cup bg\}$ indicate different spatial locations, and fg, bg are set of foreground and background spatial locations in the composite image. The segmentation network S detect the foreground region by generating foreground/background probabilities for each spatial location. We also adopt the architecture in [100] for the segmentation network.

5.3.4 Geometric and Color Consistent GAN (GCC-GAN)

Following [79], we optimize the composition model described in Equation 5.3.2 by minimizing a min-max objective:

$$min_{\theta_G} max_{\theta_D, \theta_S} \mathcal{L}_a(D, G) + \lambda \mathcal{L}_s(S, G). \tag{5.5}$$

Additional constraints are needed since directly minimizing the above objective will usually lead to the trivial solution where the compositing model simply removes the foreground in the composite image using geometric transformations. Therefore, we add a geometric loss term to our objective function:

$$\mathcal{L}_g = E_I \Big[\|T(I; \theta_G)\|_2^2 + \lambda_{mask} e^{-k \frac{\|A(I)\|_1}{N}} \Big] \tag{5.6}$$

The first term in Equation 5.6 penalizes large transformations, similar to the update loss in [87]; the second term is an exponential loss that directly penalizes the size of the foreground mask if it is too small. For data with ground-truth geometric transformation parameters, we directly use mean square error between the predict parameters and the ground-truth parameters as our geometric loss.

Finally, we use a pixel-wise L1 loss $\mathcal{L}_c$ to anchor the transformed foreground image to the original foreground image: $\mathrm{L}_c = E_I \left[\frac{\| \big(H(I_f, T(I; \theta)) - F(I) \big) \circ A(I) \|_1}{\|A(I)\|_1} \right]$.

Combining the above three loss terms, the final loss function for our GCC-

GAN becomes:

$$min_{\theta_G} max_{\theta_D} \lambda_a \mathcal{L}_a + \lambda_s \mathcal{L}_s + \lambda_g \mathcal{L}_g + \lambda_c \mathcal{L}_c, \tag{5.7}$$

where λ_a, λ_s, λ_g, and λ_c are hyper-parameters that control the weights between different loss terms.

5.4 Experiments

5.4.1 Image Compositing with Synthesized Objects

We first validate our model in a simplified artificial setting with a synthesized dataset. We first use the Panda3D game engine[1] to render images containing a table and a soda can. We render three images for each 3d configuration, including a foreground image with a soda can, a background image with a table, and a ground-truth composite image with a soda can on the table. We then apply random geometric and the color perturbation to the foreground, and we ask our model to composite the perturbed foreground into the background image. Since the synthesized images have a perfect segmentation mask, there will be no boundary artifact in the composite image. As a result, we omit the refinement network and segmentation network in our model for the experiment. We train our model on 15,000 synthesized training triplets with 200 epochs. Figure 5.3 (a) shows some example results. The first column is the initial composition with foreground perturbation, the second column is the output of our model, and the third column is the ground truth composite image. Our model is able to correct the geometric and color of the foreground and

[1] https://www.panda3d.org/

generate a plausible composite image.

Importance of Color Consistency. To demonstrate the importance of color consistency in the composite image, we also train a model with only geometric transformation network similar to the one in [87]. Figure 5.3 (b) shows the result of geometric only model. The model fails to generate plausible composite images because geometric transformation alone cannot move the composite image on to the manifold of the training data.

5.4.2 Image Compositing with Common Objects

We use Common Object in Context (COCO) [102] dataset for our compositing experiments. COCO dataset consists of 330K images with segmentation masks of 80 common object categories.

Training Data Generation. Our goal is to generate a composite image by inserting an object from a foreground image into a new background image. However, we do not have training data with realistic composite images, which requires intensive human annotation with editing software. Instead, we automatically generate training data by perturbing the input images. Figure 5.4 shows the process of training data generation. For each input image with object mask, we first select an auxiliary object mask from another image in the dataset with same object category. We then use morphological operations and combining the object mask with auxiliary mask to remove the boundaries from the image, simulating the boundary mismatch during testing. Finally, we apply the geometric and color distortion to the foreground to simulate the geometric and color mismatch during testing. For each input image I, we generate a background image I_b, a foreground image I_f and a object mask α as input to our model. Our model then tries to composite the foreground object into background and generate realistic composite image. We select object segments that occupy between 5% to 50% of the whole image for our experiments. For each segments, we select 5 auxiliary object mask with largest intersection of union, result in 516,070 training triplets. During testing, we simply remove a object from a background image, and tries to composite another foreground object with the background. Note that our goal is to evaluate image compositing algorithms, therefore we use the ground-truth object mask to segment the objects, however, we can also use semantic segmentation algorithm to segment objects for image compositing.

Compared Baselines. We compare our model with several different baselines:

- **Alpha Composition**: linear combination of the foreground and background using the alpha mask.

- **Poisson Blending** [72]: a gradient based method that minimize the gradient changes in the composite image.

- **Deep Harmonization** [78]: an end-to-end encoder-decoder network with semantic segmentation.

- **Pix2Pix** [80]: an image-to-image translation network with adversarial loss.

Figure 5.7 shows some qualitative results of the proposed method compared to baselines. Note that since the baselines do not account for geometric consistency, for fair comparison, we select a foreground object that best matches the background, and adjust the geometry to match the foreground and background mask before input to the baselines. Even without geometric mismatch, our model can automatically generate competitive or more realistic composite image compared to all baseline methods. Pix2Pix model can generate images of similar quality compared to the proposed method, however, in the following section, we show that when there is geometric inconsistency between foreground and background, Pix2Pix model will fail to generate plausible composite image since their model does not incorporate geometric losses (cf. Figure 5.6).

Importance of Geometric consistency. Figure 5.5 shows the process of geometric correction of the proposed model with some examples. The first column is the background image, the second column is the foreground object with mask. Third column shows initial composition with simple copy-paste operation. The foreground and background in the initial composition is geometrically inconsistent. In fourth column, our model first transform the foreground to make the composite image geometrically consistent using spatial transformer network. Finally, the last column shows the refinement network will make the boundary more realistic and achieve realistic image compositing. Figure 5.6 shows comparison between our model and model without geometric correction (i.e. Pix2Pix). Our model is able to perform geometric transformation to the foreground and generate plausible composite image while Pix2Pix fail to generate realistic composite image.

Human Perceptual Experiments. We conduct different perceptual experiments to quantitatively evaluate our model. In the first experiment, we want to verify

Method	**% Real**
Alpha composition	4.1%
Poisson blending [72]	10.0%
Deep harmonization [78]	8.6%
Pix2Pix [80]	10.2%
GCC-GAN (Ours)	**11.0%**
Real image	73.8%

Table 5.1: Human perceptual experiment with single image. We ask the annotator to check if there is any unusual artifact in the image. GCC-GAN can fool the annotator 11% of the time compared to baselines. Note that for fair comparison, we ensure the geometric consistency by select foreground object best matching the background.

Method	**GCC-GAN Perform Better**
Alpha composition	82.5%
Poisson Blending [72]	67.3%
Deep Harmonization [78]	71.4%
Pix2pix [80]	56.7%

Table 5.2: Human perceptual experiment with pairs of images. Given two images, we ask the annotator to select the more realistic image from pair. The output of GCC-GAN is selected more than half of the time compare to all other baselines.

how well our composite image can fool a human subject under close examination compared to baseline method. We randomly select ten images from each of the 80 categories in COCO dataset with a total of 800 images. For each image, we generate five composite images using different algorithms. We show the composite image as well as original real image to the annotator with random order and ask them to check if there is any unusual artifact in the image and obtain a total of 4,800 annotations. Table 5.1 shows the results of the experiment. Even though the input image does not require any geometric correction, our model still outperforms all baseline in term of human perception, which demonstrates the effectiveness of the adversarial training process with segmentation network. Note that 26.2% of real images were actually annotated as fake, which shows the annotator is really strict, and they annotate each image meticulously.

In the second experiments, we want to directly compare our algorithm with baselines. We randomly collect five images from each categories, with a total of 400 images. We show annotator two composite images. One image is generated by our model while the other is generated with one of the baseline methods. To ensure fair comparison, both images are generated with same foreground and background with matching object mask to ensure the composite image is geometrically consistent, and is shown to the annotator with no particular order. Table 5.2 shows the results of the experiment. Again, even without geometric correction, our model can outperform all baseline method and generate better composite image.

Qualitative results and Failure cases. Figure 5.8 show composite image generate by our model along with the original image for different object categories. Figure 5.9 show some failure cases. In the first example, our model does not have any pose information and was not able to consider semantic layout of the street scene. Therefore, the model composite the car with a inconsistent pose. In the second example, the foreground segmentation mask is imperfect (i.e. the wheel of the bike), so the model generate a composite image with inconsistent appearance. In third example, we tries to insert a color train into a black and white background, since most of our training data is color image, the model did not learn to change the appearance of foreground into black and white. In the last example, we show the failure case of composite image with an animal. Our model works better with rigid objects, and have hard time model animal with diverse poses.

Method	**Average RGB-N Score**
Alpha composition	75.4%
Poisson blending [72]	75.8%
Deep harmonization [78]	77.0%
Pix2Pix [80]	69.1%
GCC-GAN (Ours)	**63.7%**
Real image	57.8%

Table 5.3: Average manipulation score with different compositing algorithms. The score is generated by a state-of-the-art manipulation detection algorithm [1], a higher score indicates a higher possibility that an image is manipulated. GCC-GAN is able to generate more realistic images that fool the manipulation detection algorithm. Note that Poisson blending and deep harmonization perform worse than alpha composition probably because the compositing process introduces additional artifacts that capture by the manipulation detection algorithm.

Image Manipulation Detection. In this experiment, we want to see how well can composite image generated by our model fool a image manipulation detection algorithm. To this end, we utilize a well-trained state-of-the-art image manipulation detection model, RGB-N [1]. The model use a two-stream faster-rcnn network to detect different type of image manipulation. We randomly select 50 images output by each of the algorithm and pass them through the RGB-N model to generate manipulation score. Table 5.3 shows the average manipulation scores of different compositing algorithms. Our model gets lowest RGB-N score compare to all other baselines, which indicates the RGB-N model consider composite images generated by GCC-GAN are more real.

5.4.3 Implementation Details

We implement GCC-GAN with PyTorch [103] deep learning framework and train on the Nvidia GTX 1080TI GPUs. The input is resized to 128 × 128 for the experiments on synthesized dataset and 256×256 for the experiments on COCO, we use Adam [38] optimizer with a initial learning rate of 0.0002, (λ_a, λ_s, λ_g, and λ_c) are set to (0.01, 0.01, 1, 1) to empirically to balance the loss terms. We use a batch size of 1 for both experiments, and train 200 epochs for the synthesized dataset and 5 epochs for the COCO experiment. We use affine as our geometric transformation function.

5.5 Conclusion

We propose a generative network called GCC-GAN for image compositing which considers both geometric, color, and boundary consistency. We successfully use adversarial training with a discriminator network and a segmentation network to improve our model. Based on experiments on synthesized dataset as well as real world object dataset, we show both geometric and color consistency is crucial for generating realistic-looking composite images. We also GCC-GAN yield better results compare to several state-of-the-art baselines with human perceptual experiments as well as a experiment with manipulation detection algorithm. Despite the promising results, we show GCC-GAN has some limitations, such as failure to dealing with object with diverse poses. Future work includes incorporating pose information into our image compositing framework and using GCC-GAN to improve image manipulation detection algorithms.

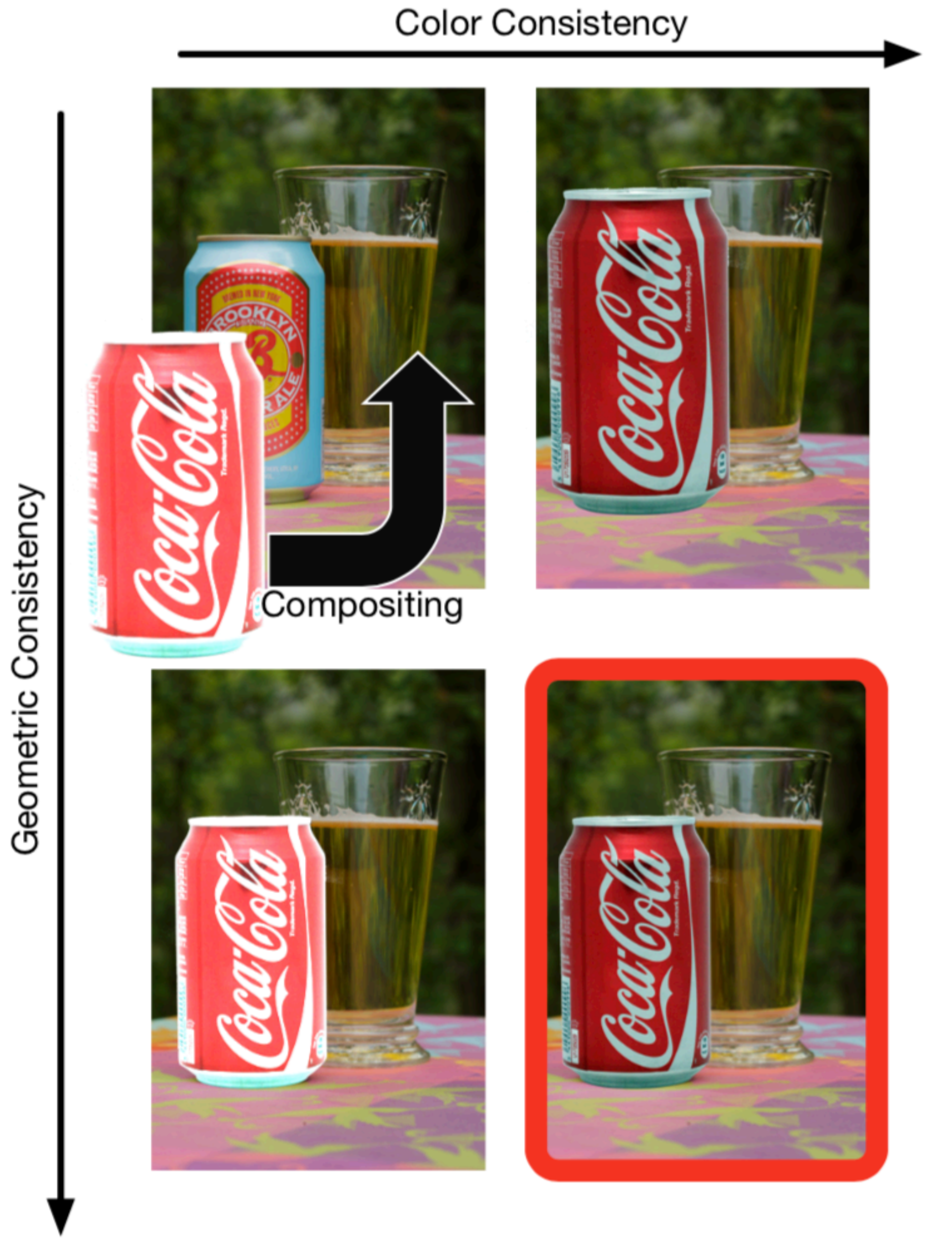


Figure 5.1: The goal of image composition is to create a realistic image by combining a foreground object with a background image. The x-axis corresponds to increasing color consistency in the composite image, while the y-axis corresponds to increasing geometric consistency. However, the composite image only looks realistic when both geometric and color consistent are considered (cf. image in the red box). (Best viewed in color)

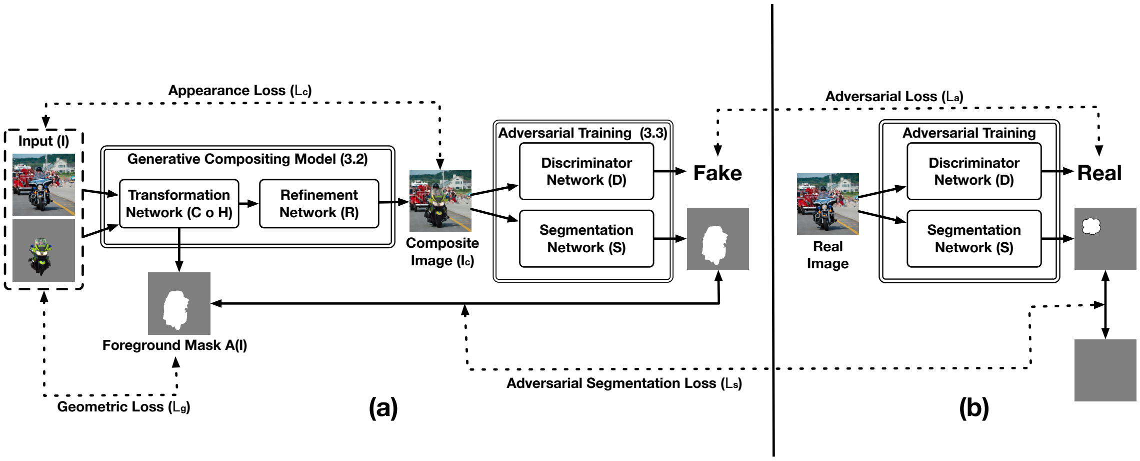


Figure 5.2: System Overview of the proposed network architecture. (a) Given an input triplet consisting of a foreground object, a foreground mask, and a background image, the generative compositing model learns to create a realistic composite image, in order to fool both the discriminator network and the segmentation network. (b) Given a real image, the discriminator network learns to predict real while the segmentation network learns to identify the image as a background.

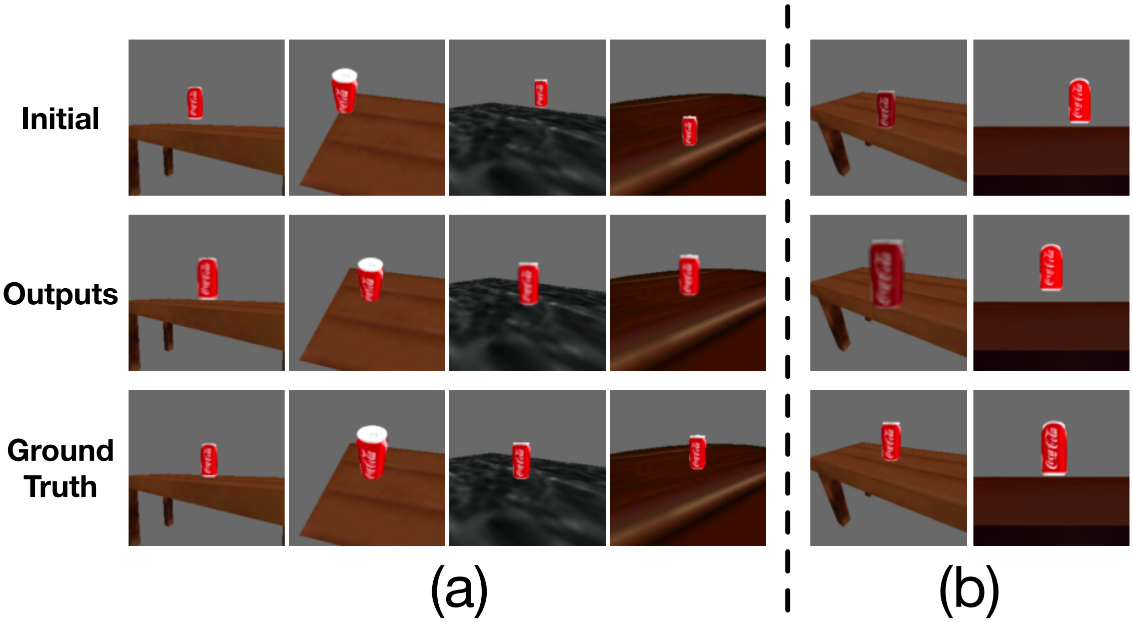


Figure 5.3: Experiments on the synthesized dataset. (a) Through geometric and color transformations, our model learns the relationship between the soda can and the table, and successfully generate composite images with a soda can sit on the table. (b) Without color transformation, the model cannot learn the correct transformation because geometric transformations alone can not move the composite image on to the manifold of the training data.

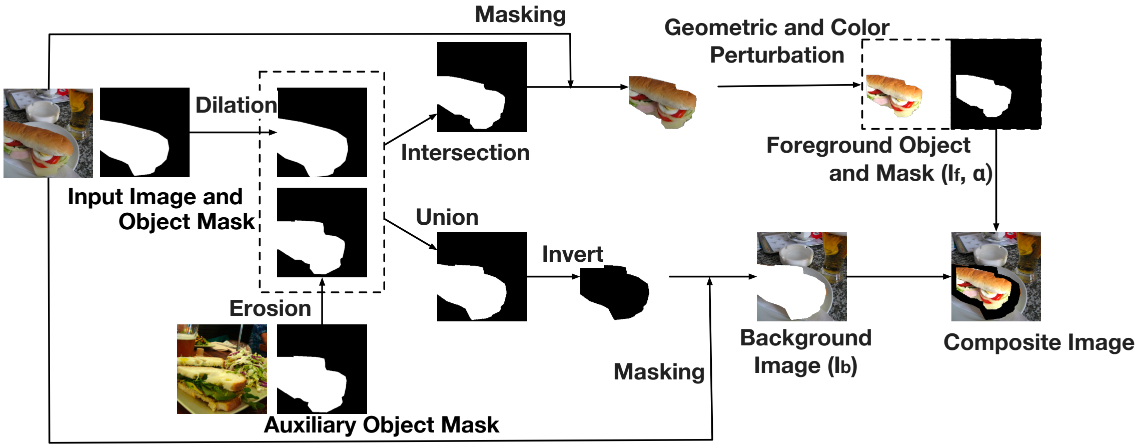


Figure 5.4: The process of training data generation. For a given image and its object mask, we first select an auxiliary object mask from a different image in the dataset with the same semantic category. We use morphological operation to remove the boundary in the foreground object and background image. We then combine the object mask with the auxiliary ones to simulate the boundary mismatch during testing. Finally, we apply geometric and color perturbation to simulate the inconsistency during testing.

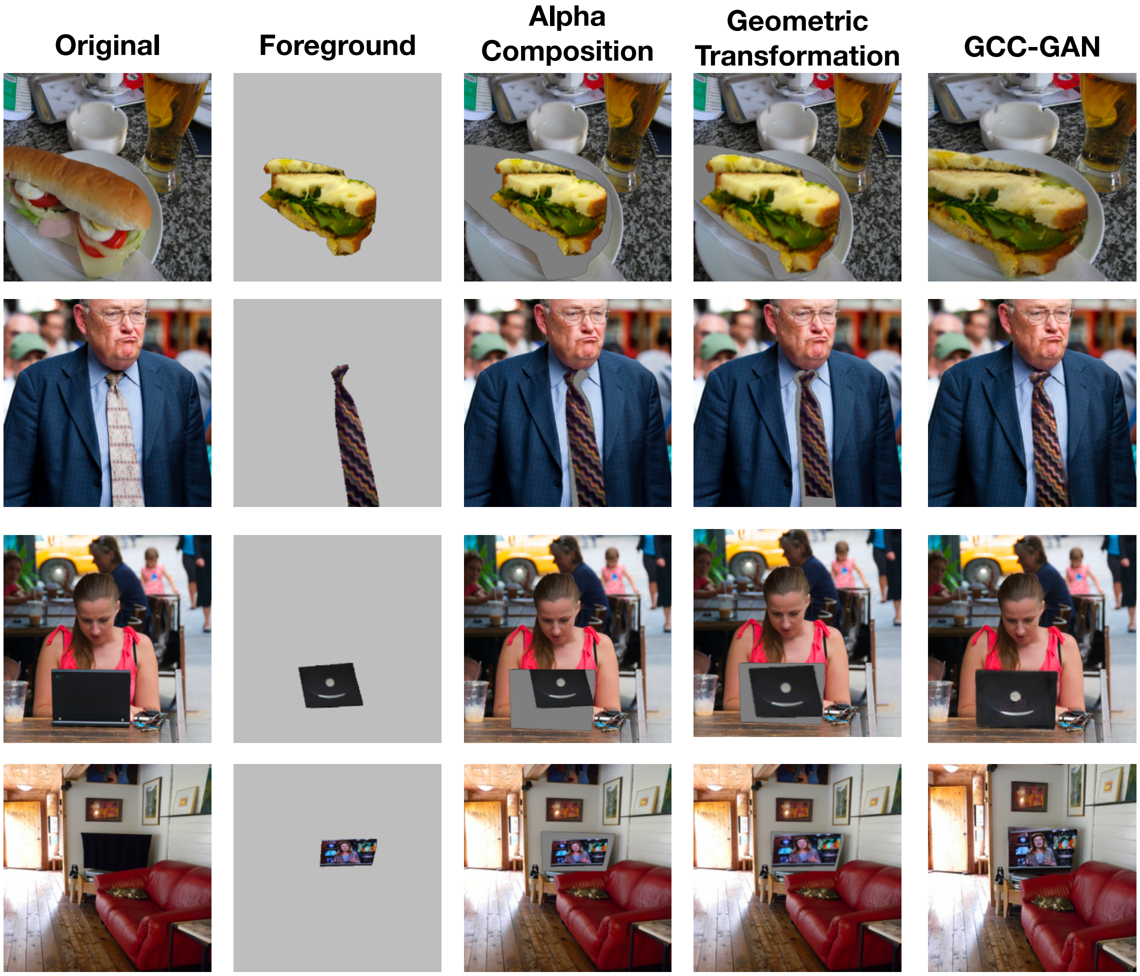


Figure 5.5: Geometric correction of GCC-GAN. The first and second column shows the original image and a foreground object. Third column shows composite image using alpha composition, the geometry is inconsistent between foreground and background. Forth column shows composite image after geometric transformation, and the last column shows the output of GCC-GAN with the final refinement network.

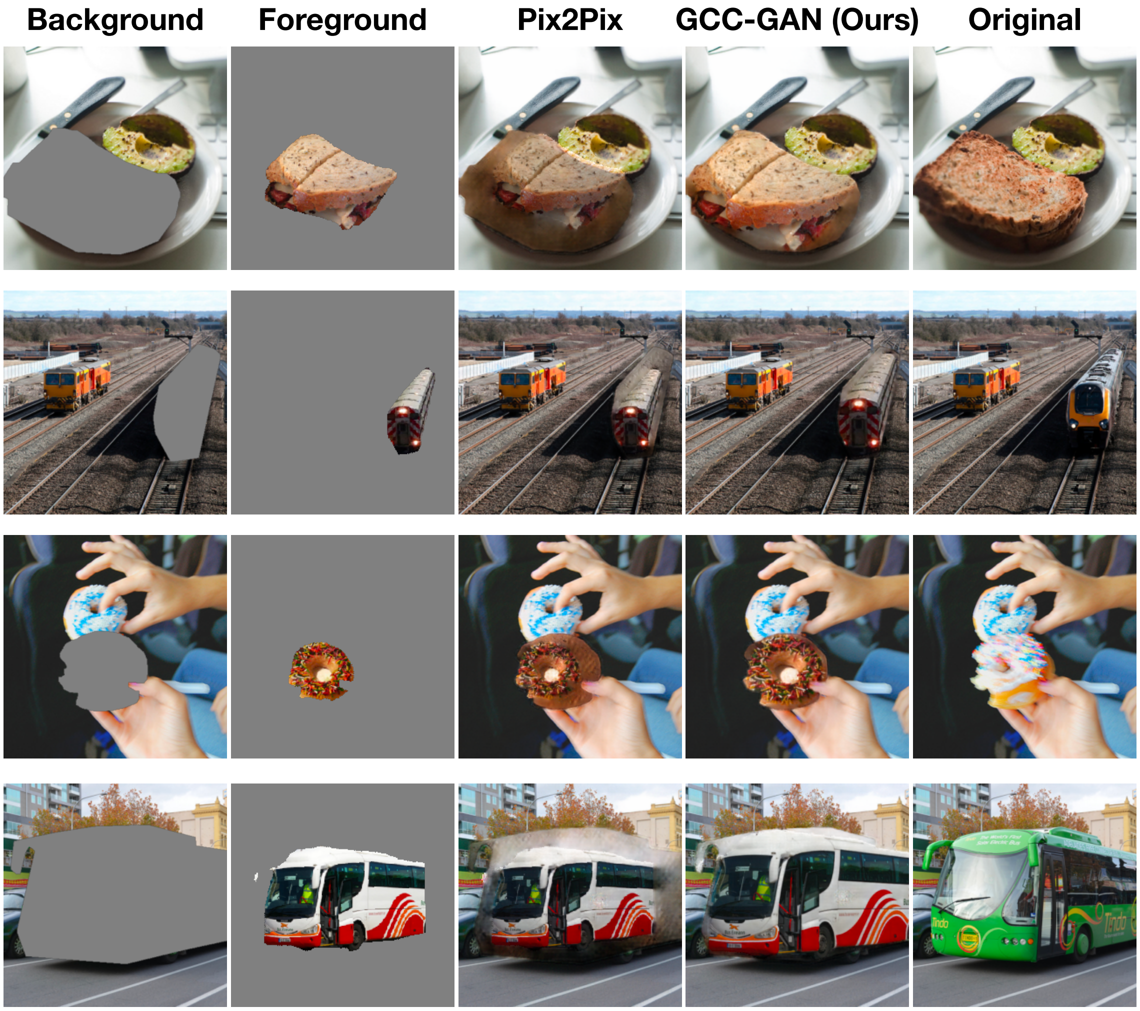


Figure 5.6: Comparison between baseline and GCC-GAN when the input geometry is inconsistent. GCC-GAN is able to correct the geometric error and generate more plausible composite image compared to baseline method (Pix2Pix).

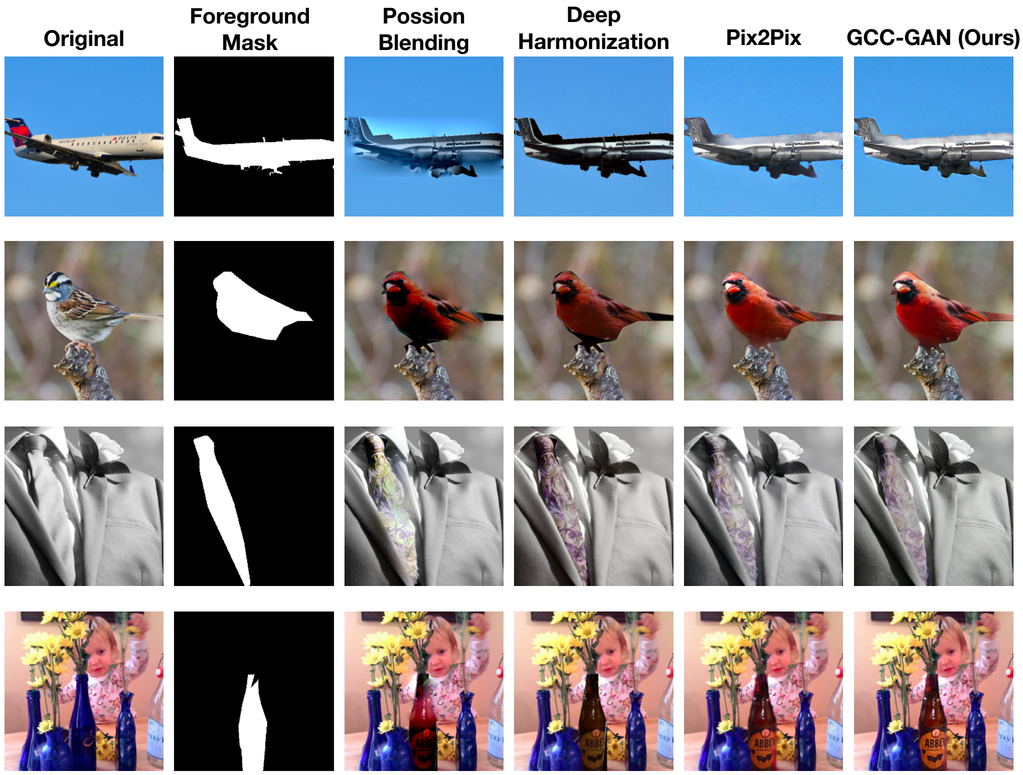


Figure 5.7: Qualitative results of different algorithms. The first column is the original image, the second column is the foreground object mask. The rest of the columns shows the outputs of different algorithms. Note that since the baseline methods do not account for geometric consistency, for fair comparison, we select foreground objects best matching the background to ensure geometric consistency.

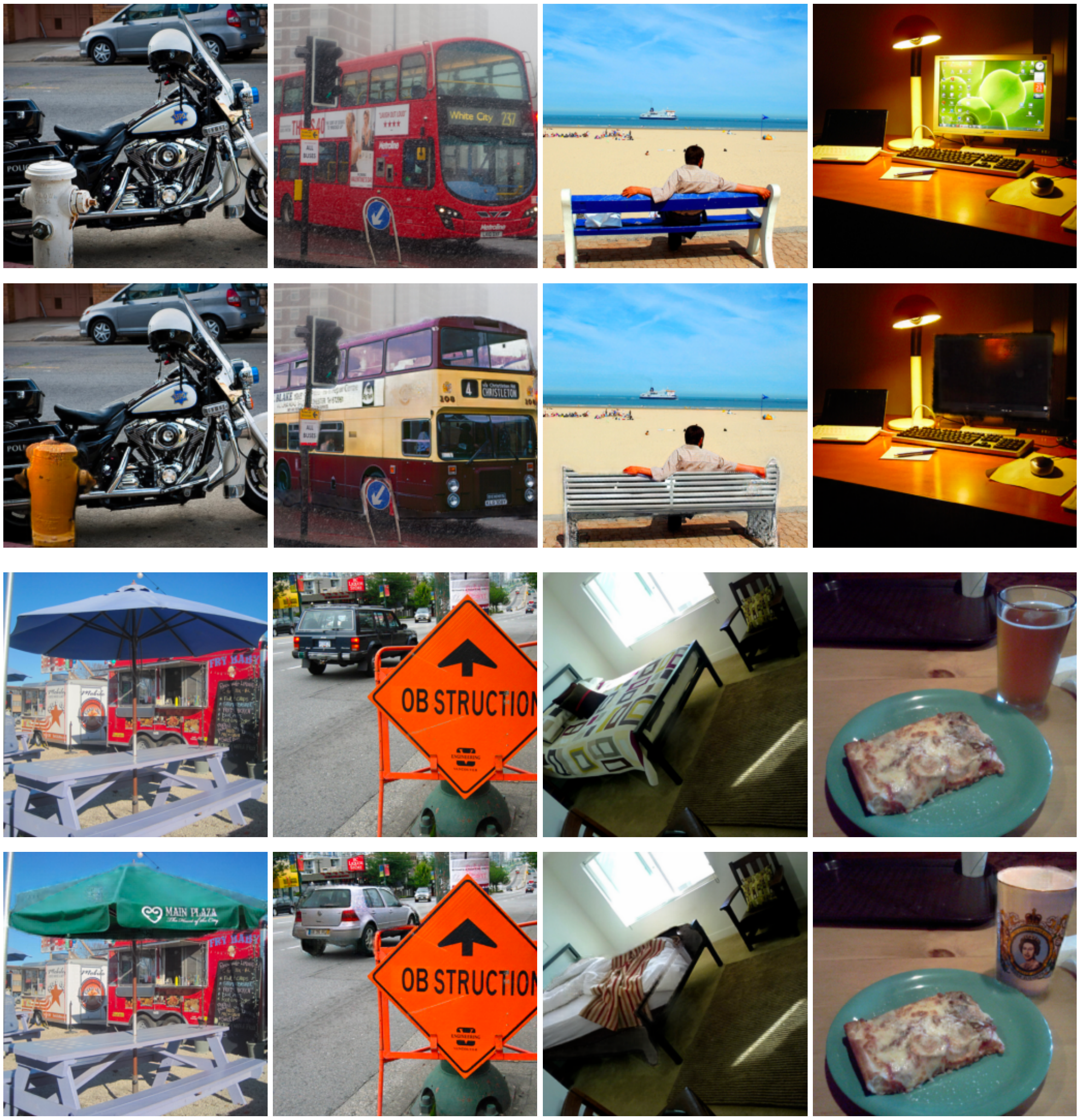


Figure 5.8: Qualitative results. The first and the third columns show the original images. The second and the fourth columns contain output of GCC-GAN.

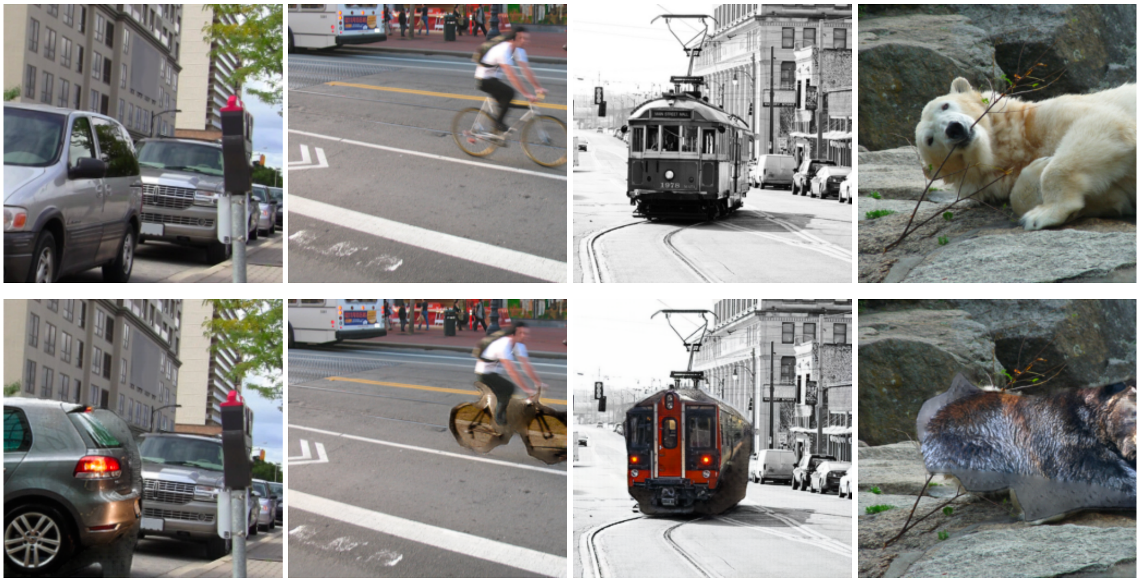


Figure 5.9: Failure cases. (1) GCC-GAN does not incorporate pose information and does not learn the semantic layout of street, therefore, composite image contain car with unrealistic pose. (2) GCC-GAN generate unrealistic image due to segmentation error. (3) Since most of our training data are color images, GCC-GAN composite a color train into a black and white background. (4) GCC-GAN perform better with rigid objects and have hard time composite object with diverse poses such as animals.

Chapter 6: An Analysis of Object Embeddings for Image Retrieval

6.1 Introduction

Convolutional neural networks trained on large-scale image classification datasets such as ImageNet [17] have been shown to be an effective generic feature extractor that can be applied to different vision tasks. These include modern object detection frameworks such as Faster-RCNN [51], which utilizes the same network architecture pre-trained on image classification datasets for feature extraction. With the availability of large-scale object detection and segmentation datasets such as COCO [102] and OpenImagesV4 [104] that come with additional bounding boxes and mask annotations, we explore whether features extracted from models trained on them would display similar effectiveness as a generic feature extractor. While ImageNet classification embeddings have been extensively studied [105, 106, 107, 108, 109, 110], little work has focused on analyzing embeddings extracted from object detection models. In this paper, we investigate the performance of such embeddings for image retrieval.

Our analysis shows that even though object detection or instance segmentation model utilizes additional annotations, the embedding learned from these models is significantly less discriminative than embeddings learn from classification models

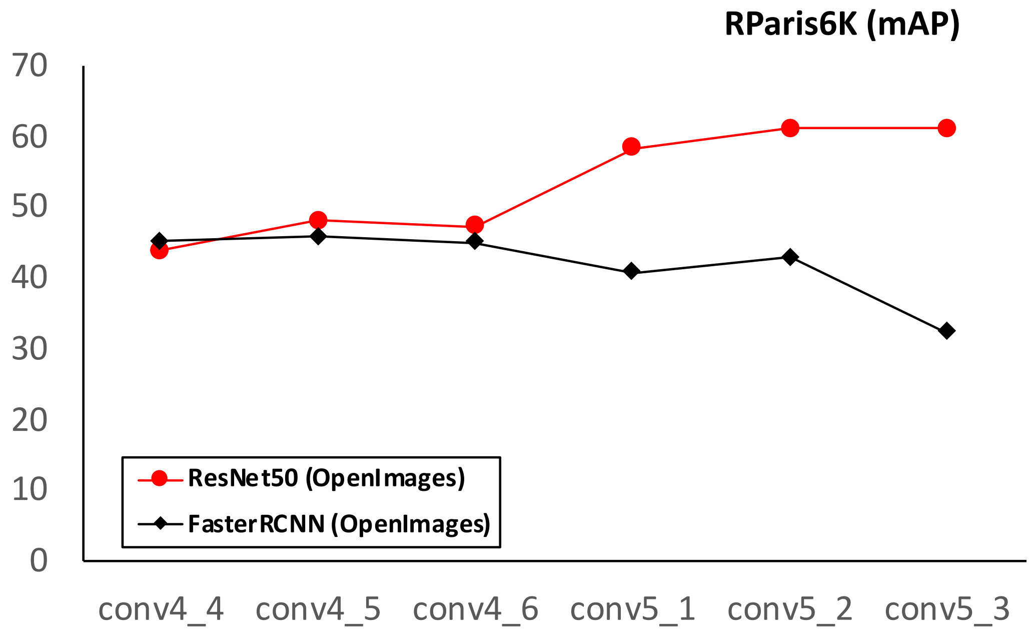


Figure 6.1: We provide a detailed analysis of embeddings extracted from different pre-trained models for image retrieval. While object detection model utilizes additional spatial annotations, embeddings extracted from the modern object detection model consistently perform worse than the classification model trained on the same dataset (OpenImagesV4) with the same backbone structure for image retrieval.

when conducting image retrieval. This suggests that the joint learning of classification and localization leads to degradation of the discriminative power of the resulting embeddings. However, we also discover that by retrieving similar objects as opposed to images, we can significantly improve image retrieval performance. For the best of both worlds, we show that by utilizing object detection as a hard attention module to extract embeddings from the classification model pertaining to the object regions, it allows the model to focus on salient regions and at the same time ignore background clutter.

For applications with an efficiency requirement, we propose a guided student-teacher training regime. We first train a teacher classification network with image-level labels as a discriminative feature extractor. This is followed by training a light-weight student network on top of the detection model that projects the feature map of the detection model into a more discriminative feature space guided by the teacher model. During image retrieval, we use the object detector as a hard attention module and extract object-level embeddings from the output of the student network with a single forward pass. This is as opposed to maintaining a separate feature extractor and an object detector, which would require two forward passes. Such a student network would still decouple feature learning from localization, which helps to preserve the discriminative power of the features. It is also possible to learn different student transformations without re-training the object detection model.

Our contributions include: (1) We empirically show that embeddings extracted from object detection models are less discriminative than embeddings extracted from image classification models when the task of image retrieval is considered. (2) We demonstrate that an object detector can help image retrieval performance by acting as a hard attention module. (3) For efficiency, we propose a student-teacher training paradigm, which allows us to extract discriminative object embeddings in a single forward pass. (4) Finally, extensive experimental results show the advantage of the proposed approach. Further, we also demonstrate the efficacy of our approach for near-duplicate object retrieval, which allows for an important application in detecting image splicing.

6.2 Background and Related Work

Representation learning from large-scale datasets. Previous works mainly studied the transferability of embeddings extracted from classification models that have been trained on datasets such as ImageNet to other tasks [105, 106, 107, 108, 109, 110]. For instance, [106] reports comprehensive results of applying embeddings from the ImageNet-trained classification model to object detection, scene recognition, as well as image retrieval. In contrast, the efficacy of embeddings obtained from object detection models trained on large-scale object detection datasets such as COCO [102] and OpenImages [104] has not been widely studied. In this work, we provide an analysis of embeddings extracted from different models pre-trained on large-scale datasets for the retrieval task.

Content-based image retrieval aims to retrieve relevant images from an image database given a query image based on the image content. Early work [111] used global color and texture statistics such as color histogram and Gabor wavelet transform to represent the image. Later advances on instance retrieval using local feature [7] and indexing methods [112, 113, 114] achieved robustness against illumination and geometric variations. With the recent broad adoption of convolutional neural networks (CNN), different techniques has been proposed for global feature extraction [115, 116, 117, 118, 119, 120], local feature extraction [121, 122, 123], embedding learning [124, 125, 126, 127], as well as geometric alignment [128, 129, 130] using deep networks. Zheng [131] provide a comprehensive review of recent approaches towards image retrieval. Different from traditional image retrieval using either global features or local features, our approach generates a few discriminative object embeddings utilizing object detection models for image retrieval.

Object detection aims to detect different objects in an input image. Girshick [132] proposed one of the first deep learning based object detection models, R-CNN, which improved the accuracy significantly compared to traditional methods [133, 134, 135]. Since then many enhancements [51, 136, 137, 138] have been made to improve accuracy as well as the training/inference time. A comprehensive survey of recent deep learning based object detection methods can be found in [139]. By taking advantage of recent success in object detection, our model can learn discriminative object-level embeddings for image retrieval. Most recently, Teichmann [140] utilized a specialized landmark detection model to aggregate deep local features [121] for landmark retrieval. Object detection has also been used to improve the performance of other vision tasks such as visual question answering [141].

Knowledge distillation [142, 143, 144, 145, 146] compress a complex model into a simpler one while maintaining the accuracy of the model. Bucilua [142] first proposed to train a single model to mimic the outputs of an ensemble of models. Ba [143] adopted a similar idea to compress deep neural networks. Hinton [144] further generalized the idea with temperature cross-entropy loss. Our student-teacher approach is related to knowledge distillation, which learns a simple student model to mimic the output of a complex one. What is different is that we leverage a detection network to provide additional guidance during training, which we show is effective for training the student network.

Model	Training Set (# of Img. / Cls.)	$\mathcal{R}$Oxf	$\mathcal{R}$Par	CUB200	Cars196
Faster-RCNN [51]	COCO (330K / 80)	18.7	28.3	4.1	2.4
Faster-RCNN-FPN [137]		20.4	31.0	3.3	3.1
Mask-RCNN [147]		20.7	33.0	3.0	2.4
Mask-RCNN-FPN [137]		34.2	48.1	2.9	3.6
ResNet50 [37]	ImageNet (1.2M / 1K)	**40.1**	**57.3**	**21.2**	**11.1**
Faster-RCNN [51]	OpenImagesV4 (1.7M / 601)	19.5	32.3	4.7	2.2
ResNet50 [37]		**41.2**	**61.2**	**19.3**	**11.0**

Table 6.1: Image retrieval performance (mAP) with embeddings extracted from different pre-trained models for four different retrieval benchmarks. Even though all detection and instance segmentation models are initialized with weights trained on ImageNet classification dataset, the embeddings learned from these models perform significantly worse than embeddings learned from the classification model.

6.3 Analyzing Embeddings for Image Retrieval

6.3.1 Embeddings from Pre-trained Models

We first provide a detailed analysis of embeddings extracted from different pre-trained models, including image classification, object detection and instance segmentation models using four different retrieval benchmarks.

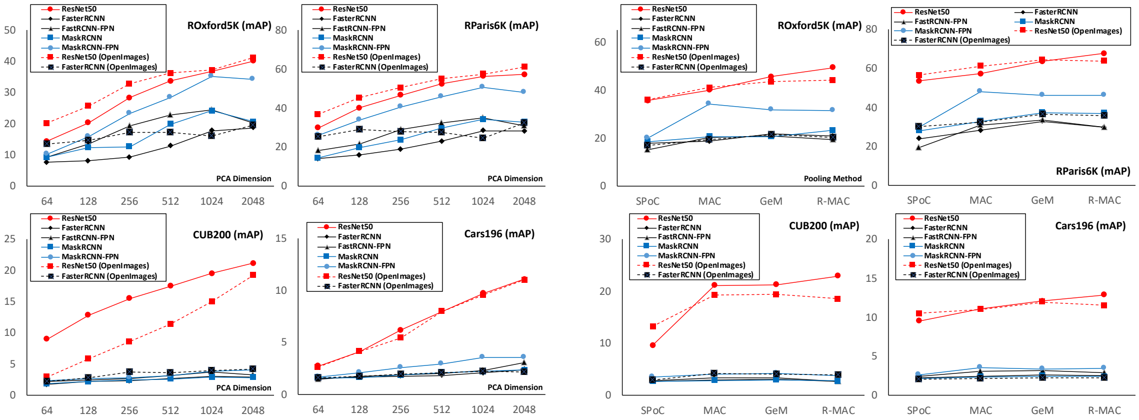
Retrieval benchmark. We consider four datasets for benchmarking, including USCB bird dataset [148] (**CUB200**), Stanford car dataset [149] (**Cars196**), and two landmark datasets, $\mathcal{R}$Oxford5K [150] (**$\mathcal{R}$Oxf**) and $\mathcal{R}$Pairs6K [150] (**$\mathcal{R}$Par**). For CUB200 and Cars196 we follow the same protocol in [151] and use leave-one-out partitions to evaluate on every images in the test set. For $\mathcal{R}$Oxford5K and $\mathcal{R}$Paris6K we follow the medium protocol described in [150], using 70 and 55 images as queries, 4,993 and 6,322 images as database. We use mean average precision (mAP) to measure the performance of different embeddings.

Pre-trained models. We consider seven different pre-trained models including (1) **Faster-RCNN** [51] and (2) Faster-RCNN with feature pyramid networks [137] (**Faster-RCNN-FPN**) trained on COCO [102], (3) **Mask-RCNN** [147] and (4) Mask-RCNN with feature pyramid networks (**Mask-RCNN-FPN**) trained on COCO with bounding box and mask annotations, and (5) **ResNet50** [37] trained on ImageNet. To control the effect of different training data, we also compare with (6) **Faster-RCNN** and (7) **ResNet50** trained with the same dataset (OpenImagesV4 [104]). We adopt open source implementation[1] of Faster-RCNN and Mask-RCNN with ResNet50 as a backbone feature extractor for all our detection and segmentation models and the same backbone as our classification model. For all Faster-RCNN and Mask-RCNN models, we use weights from the ImageNet classification model to initialize the backbone network and use the default 3x learning rate schedule to train the models. We use images from OpenImagesV4 to learn project matrix for PCA dimensionality reduction.

During test time, we first resized the image to a maximum size of 1024×1024, we then extract features from conv5_3 layer [37] and used max-pooling to produce image embeddings from different pre-trained models. We then use cosine similarity between embeddings for retrieval ranking. Note that for a fair comparison, we do not apply any post-processing tricks such as multi-scale ensemble and query expansion.

Embeddings comparison. Table 6.1 shows the mean average precision of different models when used as feature extractors on the four retrieval benchmarks. Comparing Faster-RCNN (COCO) and Mask-RCNN (COCO), we note that addi-

[1] https://github.com/facebookresearch/detectron2



(a) PCA Dimension

(b) Pooling

Figure 6.2: Analysis of embeddings with (a) different PCA dimension and (b) different pooling techniques. Embeddings learned from classification model consistently achieve the best performance.

tional mask annotations decrease the performance of the embeddings on some of the dataset, suggesting that additional localization constraints might even hurt the retrieval performance further. Also, by increasing the size of the training set from COCO to OpenImagesV4, the Faster-RCNN performance improves on some datasets but degrades on other datasets. Most importantly, although all the models are initialized with weights trained on ImageNet classification, embeddings extracted from detection and segmentation models perform significantly worse than the embeddings from the ImageNet classification model. Even when comparing Faster-RCNN (OpenImages) with ResNet50 (OpenImages) which are trained with the same training data, but with Faster-RCNN utilizing more human annotations (i.e. bounding boxes), embeddings learned from classification model still significantly outperform embeddings learned from detection model. This suggests that enforcing both classification and localization during training compromises the discriminative ability of the embedding. Consequently, decoupling localization and classification might be

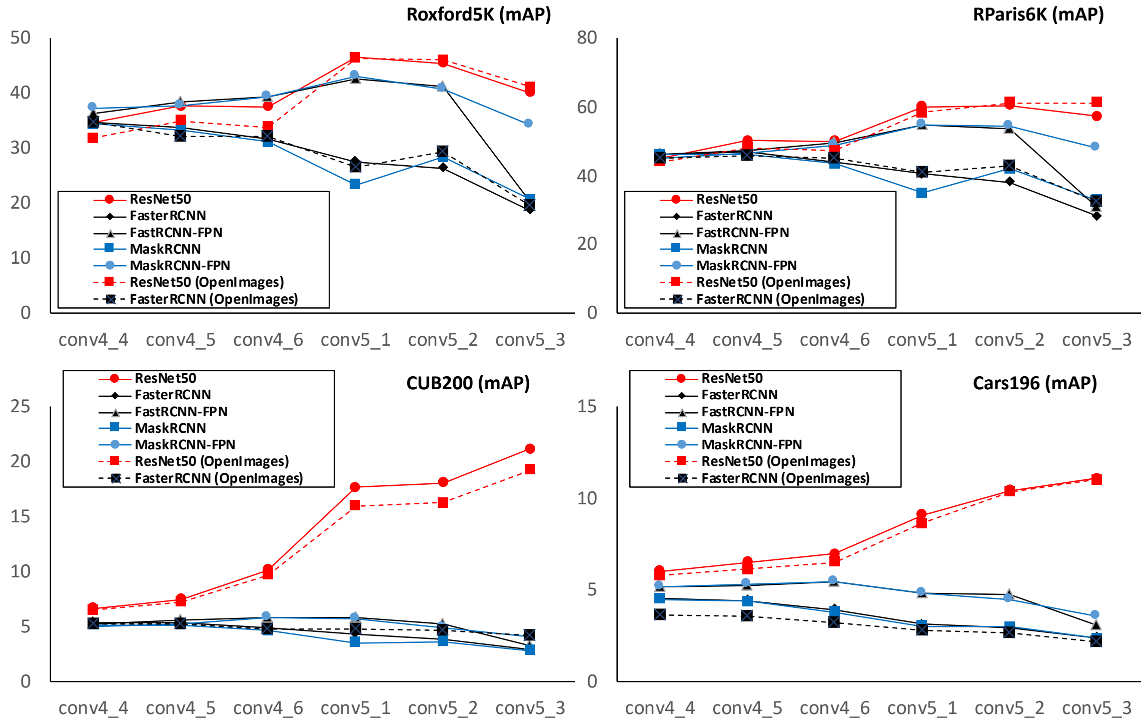


Figure 6.3: Performance of embeddings extracted from different layers of the pre-trained models. Embeddings from lower layers of classification and detection models have similar performance as they learn similar low-level texture features. However, their performance starts to diverge as we use higher layers, with the classification model achieving better performance.

crucial for learning embeddings that are effective for image retrieval.

PCA and pooling. Note that different spatial pooling techniques [150] and post-processing steps such as dimensionality reduction [152] have been shown to greatly affect retrieval performance. Given a convolutional feature map from conv5_3 layer $F \in R^{W \times H \times C}$, we consider the following pooling functions $P: R^{W \times H \times C} \to R^{C}$: (1) sum pooling [116] (SPoC), (2) max-pooling [153] (Max), (3) regional max-pooling [117] (R-MAC), and (4) generalized mean pooling [154] (GeM). We also perform experiments while varying the number of dimensions in PCA from 64 to 2,048 with whitening. Figure 6.2 shows a detailed analysis of the effect of different pooling techniques and post-processing steps. Figure 6.2 (a) shows retrieval performance of four benchmarks with different PCA dimensions. Even though the performance of all embeddings decreases as the feature dimension goes down, embeddings from the classification model (ResNet50) consistently perform the best for all dimensions, which further supports our previous observation. Figure 6.2 (b) shows the mAP for different pooling techniques. Here, ResNet50 embeddings again consistently achieve the best performance among embeddings from different pre-trained models on all datasets.

Embeddings from different layers. Figure 6.3 shows the performance with embeddings extracted from different layers in ResNet50 backbone from conv4_1 to conv5_3. Note that for lower-level embeddings, detection models and classification models share similar performance, because they represent similar low-level texture features. However, their performance diverges for embeddings from high-level layers. This is an important observation since embeddings extracted from higher level (conv5_x) achieve better retrieval performance across all datasets. This again supports the embeddings from classification models as being better suited for image retrieval.

Unsupervised clustering. To provide additional evidence that image classification model learns better embeddings compare to detection models, we conduct additional experiments by performing k-means clustering using embeddings extracted from different pre-trained models, and evaluate the cluster quality based on normalized mutual information (NMI). As shown in Table 6.3, embeddings extracted from the image classification model achieve better clustering results in terms

Dataset	$\mathcal{R}$Oxf		$\mathcal{R}$Par	
Embeddings	mAP	P@10	mAP	P@10
CNN	40.1	61.3	57.3	96.7
CNN-OE	**53.4**	**76.0**	**69.7**	**98.6**

Table 6.2: Performance of image retrieval by utilizing object detection model. We use object detection as a hard attention module for extracting object-level regional embeddings from convolutional feature maps for image retrieval. Retrieval performance in terms of mean average precision (mAP) and precision at ten (p@10) both shows significant improvement compared to using a single embedding from the whole image.

Model	(Training Set)	CUB200	Cars196
Faster-RCNN	COCO	25.5%	21.3%
Faster-RCNN-FPN		26.0%	23.3%
Mask-RCNN		24.7%	20.8%
Mask-RCNN-FPN		29.2%	24.7%
ResNet50	ImageNet	**57.6%**	**39.0%**
Faster-RCNN	OpenImagesV4	30.8%	20.4%
ResNet50		**55.8%**	**41.0%**

Table 6.3: NMI of embeddings from different models. Similar to the results of image retrieval, embeddings from the classification model also show superior performance compared to features from the detection models.

of NMI compared to the detection model and segmentation model. This demonstrates that embeddings from the classification model are better suited for both image retrieval as well as unsupervised clustering task.

6.3.2 Can Object Detection Help Image Retrieval?

Even though the embeddings extracted from object detection models are less discriminative, here we show how localization can be beneficial when conducting image retrieval. Using the same benchmarks, we show that by explicitly utilizing object

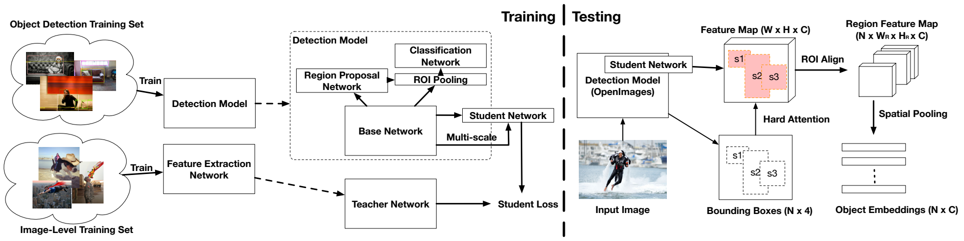


Figure 6.4: Overview of the student-teacher training paradigm. We first train a teacher classification network to learn discriminative features, and a separate object detection model for bounding box prediction. Finally, we train a compact student network to transform the feature map from the detection model to the discriminative feature space, guided by the teacher model.

bounding boxes predicted by the detection model as a hard attention mechanism, thereby ignoring background clutter, image retrieval performance can be improved. Specifically, for each image, we first deploy the object detection model trained on the OpenImagesV4 dataset to detect up to eight bounding boxes per image. For each bounding box, object-level embedding is extracted from conv5_3 layer (with resolution up to 32×32) of ResNet50 model pre-trained on ImageNet using an ROI align layer [147]. To compute the similarity between two images, we first aggregate the convolutional feature map with max-pooling and compute the maximum similarity between pairwise objects embeddings. Table 6.2 shows mAP and precision at ten (P@10) of image retrieval when using the image embeddings (**CNN**) and the object-level embeddings (**CNN-OE**). CNN-OE achieves better performance across different datasets, which suggests the detection model can help retrieval by acting as a hard attention mechanism.

6.4 Efficient Image Retrieval using Object Embeddings

Section 6.3.2 provides a simple approach toward utilizing object detection for improving retrieval performance. However, CNN-OE uses two separate models: a classification model used for generating discriminative feature maps, and a detection model responsible for the hard attention, resulting in two forward passes during inference. To be more efficient, we propose to use knowledge distillation [144] to combine the two models. Figure 6.4 shows the overview of our approach for image retrieval. During training, we first train a classification teacher model that learns to generate discriminative features as well as a separate object detection model. We then train a student network that transforms the feature map from the object detection model to the teacher model. During test time, the combined model outputs both the bounding box predictions as well as the discriminative feature maps. ROI align layer with spatial pooling is used to extract object embeddings from the feature maps to perform retrieval.

Training student networks. Figure 6.5 illustrates three different types of student networks. We first consider a simple model compression approach by training a compact student model to directly mimic the output of the teacher network (cf. Figure 6.5 (a)). Given an input image I, a pre-trained teacher network $T\colon R^{W \times H \times 3} \rightarrow R^{\frac{W}{32} \times \frac{H}{32} \times C}$, we construct a student network $S_{full}\colon R^{W \times H \times 3} \rightarrow R^{\frac{W}{32} \times \frac{H}{32} \times C}$ with one convolutional layers and four bottleneck layers with skip connections [37] and parameters θ_s. We directly minimize the mean squared error between

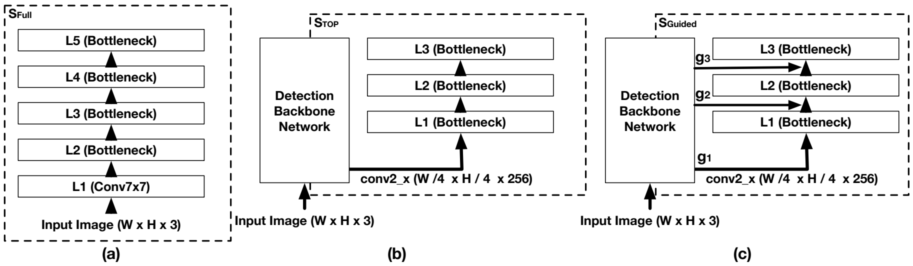


Figure 6.5: Three different types of student networks. (a) Compact student network S_{full} that directly takes input images and tries to mimic the output of the teacher network. This can be considered as a simple model compression approach. (b) Student network that utilizes the low-level features from the detection model S_{top}; it is more compact compared to S_{full}, since it reuses the lower layers from the detection model. (c) Student network with multi-scale guidance S_{guided}. It takes both high-level and low-level feature maps from the detection model as guidance to learn the discriminative features from teacher network.

the output feature maps using gradient decent:

$$minimize_{\theta_s} \sum_{I} ||S_{full}(I;\theta_s) - T(I)||_2. \tag{6.1}$$

It is commonly believed that the shallower layers in convolutional neural networks learn common low-level features such as edges which can be useful for all visual tasks. Since we already compute these low-level features in the detection model, we can reuse them for training the student model. The detection model's backbone network is represented as $(D_{l4} \circ D_{l3} \circ D_{lower})(\cdot)$, where $D_{lower}\colon R^{W \times H \times 3} \to R^{\frac{W}{4} \times \frac{H}{4} \times \frac{C}{8}}$, denotes the lower layers in the network; $D_{l3}\colon R^{\frac{W}{4} \times \frac{H}{4} \times \frac{C}{8}} \to R^{\frac{W}{8} \times \frac{H}{8} \times \frac{C}{4}}$, and $D_{l4}\colon R^{\frac{W}{8} \times \frac{H}{8} \times \frac{C}{4}} \to R^{\frac{W}{16} \times \frac{H}{16} \times \frac{C}{2}}$ are the higher layers. We consider a student model $S_{top}\colon R^{\frac{W}{4} \times \frac{H}{4} \times \frac{C}{8}} \to R^{\frac{W}{32} \times \frac{H}{32} \times C}$ that only contains the top layers (cf. Figure 6.5 (b)). Reusing the lower layers from the detection network $D_{lower}\colon R^{W \times H \times 3} \to R^{\frac{W}{4} \times \frac{H}{4} \times \frac{C}{8}}$, the mean squared error between the output feature maps is minimized:

$$minimize_{\theta_s} \sum_I ||S_{top}(D_{lower}(I); \theta_s) - T(I)||_2. \tag{6.2}$$

Lastly, we propose a guided student model S_{guided}: $(R^{\frac{W}{4}\times\frac{H}{4}\times\frac{C}{8}}, R^{\frac{W}{8}\times\frac{H}{8}\times\frac{C}{4}}, R^{\frac{W}{16}\times\frac{H}{16}\times\frac{C}{2}}) \rightarrow R^{\frac{W}{32}\times\frac{H}{32}\times C}$ that uses multi-scale feature maps from the detection backbone network as guidance to learn discriminative embeddings (cf. Figure 6.5 (c)), with each layer L_i of S_{guided} defined as:

$$y_i = L_i(y_{i-1} + g_{i-1}), \tag{6.3}$$

where L_i is a bottleneck layer, y_i is the output of layer i, and g_1, g_2, g_3 are the guidance inputs from the detection backbone network with $y_0 = g_1$ and $g_0 = 0$. Here, we assume the guidance has the same dimension as the layer output of the student model. For different dimensions, a linear transformation is applied to map them into the same space. Finally, we minimize the mean squared error between the output of the student model S_{guided} and the teacher model T: minimize$_{\theta_s} \sum_I ||S_{guided}(g_1, g_2, g_3; \theta_s) - T(I)||_2$. Student model with multi-scale guidance can utilize both high-level and low-level features learned in the detection model. As shown in Section 6.5.1, this is essential for learning discriminative features.

Embeddings	FLOPs	# Params.	$\mathcal{R}$Oxf	$\mathcal{R}$Par
Faster-RCNN	-	-	25.4	34.4
Student - Full (S_{full})	1.49×10^9	8.02×10^6	32.1	55.2
Student - Top (S_{top})	1.13×10^9	7.93×10^6	43.3	56.3
Student - Guided (S_{guided})	$\mathbf{0.82 \times 10^9}$	$\mathbf{5.17 \times 10^6}$	**50.2**	**65.2**
Teacher (CNN-OE)	3.33×10^9	8.54×10^6	53.4	69.7

Table 6.4: FLOPs, number of parameters and mAP for different student models. The performance of the proposed S_{guided} achieves better performance while using fewer FLOPs and model parameters comparing to two other baseline student model.

6.5 Experimental Details

6.5.1 Experiment with Different Student Networks

We use images from the OpenImageV4 dataset to train different student models. Note that the training of the student model is unsupervised and does not require any manual annotations. We use Adam [38] optimizer with a learning rate of 1-e3 and batch size of 64 to train all the student models for 20,000 iterations. Table 6.4 shows the performance of different student models in terms of mAP. S_{full} achieves the worst performance and it struggles to learn discriminative embeddings. S_{top} achieves slightly better performance than S_{full} by reusing the low-level feature maps from the detection model. Utilizing the guidance from multi-scale feature maps of the detection model, our guided student model S_{guided} obtains the best performance. Note that the proposed guided student model actually also requires the least amount of computation, with only one-fourth of the FLOPs used by the teacher model while obtaining up to 93.6% of the performance.

6.5.2 Experiments on Landmark Retrieval

Table 6.5 (top) compares different landmark retrieval approaches with ImageNet pretrained models, including sum pooling of convolutions (**CNN-SPoC**) [116], maximum activation of convolutions (**CNN-MAC**) [153], regional maximum activation (**CNN-R-MAC**) [117], and generalized mean pooling (**CNN-GeM**) [154] on $\mathcal{R}$Oxford5K and $\mathcal{R}$Paris6K dataset. For a fair comparison, we employ the same ResNet50 pre-trained on ImageNet for all the methods. Also, we do not apply any additional post-processing except PCA whitening. Our approach (CNN-OE) achieves the best performance among other approaches using the same pre-trained network; in addition, our approach still maintains competitive results using the compact student network (CNN-OE-S_{guided}) described in Section 6.4. Table 6.5 (bottom) compares different state-of-the-art approaches on the same dataset. Note that state-of-the-art methods utilize different additional training data. For example, Radenovic [120] utilize training data pairs collected from spatial verification with local features while Teichmann [140] utilize Google landmark dataset as additional training data. Here we also utilize the Google landmark dataset to fine-tune our model and extract object embeddings from the fine-tuned model. Details on the training process are described in the supplementary material. Our model (CNN-FT-OE) achieves state-of-the-art performance without any post-processing except PCA whitening and also achieves competitive performance compared to the model that uses re-ranking techniques such as spatial verification.

Method	$\mathcal{R}$Oxford5K		$\mathcal{R}$Paris6K	
w/ ImageNet pretrained model	mAP	P@10	mAP	P@10
CNN-SPoC [116]	35.7	55.4	53.5	90.3
CNN-MAC [153]	40.1	61.3	57.3	96.7
CNN-R-MAC[117]	49.4	70.4	**67.6**	98.1
CNN-GeM [154]	45.7	67.2	63.6	96.3
CNN-OE-S_{guided} (Ours)	**50.2**	**71.2**	65.2	**98.1**
CNN-OE (Ours)	**53.4**	**76.0**	**69.7**	**98.6**
w/ additional training data	mAP	P@10	mAP	P@10
ResNet101-R-MAC [120]	60.9	78.1	78.9	96.9
ResNet101-GeM [119]	64.7	84.7	77.2	98.1
DELF–D2R-R-ASMK [140]	73.3	90.0	80.7	99.1
DELF–D2R-R-ASMK+SP [140]	76.0	**93.4**	80.2	**99.1**
CNN-FT-OE (Ours)	**78.7**	91.8	**83.4**	98.3

Table 6.5: Comparison of different approaches on $\mathcal{R}$Oxford5K and $\mathcal{R}$Paris6K datasets with or without additional training data. Our approach achieves the best performance among other baselines even when a compact student model is deployed. For model with additional training data, our model achieves competitive performance even when comparing with the model using a re-ranking method such as spatial verification.

6.5.3 Fine-Grained Image Retrieval

Table 6.6 compares the proposed method with state-of-the-art embedding learning approaches on two fine-grained image datasets, CUB200, and Cars196. We compare several embedding learning approaches including **ProxyNCA** [124], **Angular Loss** [126], **Margin Loss** [125], **Hierarchical Triplet Loss (HTL)** [127], and Multi-Similarity Loss **Multi-Sim**. Note that it is hard to fairly compare different methods as they use different network architecture or embedding dimension. Nevertheless, we show the precision at one (P@1) of the proposed method to provide insights into how it compares with the state-of-the-art.

Ablation study. By using object embeddings from ImageNet pretrained model (**CNN-OE**), we can achieve 61.02% precision on CUB200, which is already quite competitive with the state-of-the-art embedding learning approach. To ensure that performance gain does not just come from using more descriptors for one image, we also provide a baseline approach that randomly samples the same number of bounding boxes from the images to extract embeddings (**CNN-RandomBoxes**). Results show that CNN-RandomBoxes performs worse than CNN-OE, which demonstrates the importance of utilizing object detector as a hard attention mechanism. For a fair comparison, we note that the SOTA methods have all been trained on the training sets of CUB200 and Cars196, while CNN-OE is simply using the weights of the ImageNet classification model. For this reason, we also fine-tune the classification model with the training set corresponding to each benchmark (**CNN-FT-OE**) using the same training process as the experiment on landmark retrieval and us-

Method	Network	Dimension	CUB200		Cars196	
			mAP	P@1	mAP	P@1
ProxyNCA [124]	Inception BN [155]	64	-	49.2	-	73.2
Angular Loss[126]	GoogLeNet [156]	512	-	54.7	-	71.4
Margin Loss [125]	ResNet50 [37]	128	-	63.6	-	79.6
HTL [127]	Inception BN [155]	512	-	57.1	-	81.4
Multi-Sim [157]	Inception BN [155]	512	-	65.7	-	84.1
CNN-OE	ResNet50 [37]	2048	23.8	61.0	12.1	61.9
CNN-RandomBoxes		2048	21.8	58.1	10.0	53.3
CNN-OE + PCA		512	22.6	58.0	10.0	54.4
CNN-FT + PCA		512	31.2	62.0	26.0	82.3
CNN-FT-OE + PCA		512	**32.0**	**66.5**	**28.8**	**84.3**

Table 6.6: Comparison with state-of-the-art approaches on CUB200 and Cars196. Object embeddings from ImageNet pre-trained model (CNN-OE) obtain competitive results on CUB200. By fine-tuning on the training set corresponding to each benchmark (CNN-FT-OE), we can achieve state-of-the-art retrieval performance.

Method	COCO-Fake		PIR	
	mAP	P@10	mAP	P@10
CNN-SPoC [116]	32.3	34.3	40.8	61.7
CNN-MAC [153]	32.8	34.8	46.9	68.1
CNN-R-MAC[117]	40.9	42.5	44.2	65.7
CNN-GeM [154]	41.3	43.7	44.4	65.5
CNN-OE (Ours)	**82.1**	**82.9**	**54.1**	**75.0**

Table 6.7: Performance on PhotoShop Image Retrieval (PIR) dataset. Our approach is especially suitable for retrieving tampered images with spliced objects.

ing PCA to reduce the dimension of the embeddings to 512. With fine-tuning, our approach (CNN-FT-OE + PCA) can achieve the state-of-the-art performance of 66.48% on CUB200 and 84.27% on Cars196.

6.5.4 Near-Duplicate Object Retrieval

One interesting capability of our proposed approach is in retrieving near-duplicate objects in images. Having demonstrated that our approach works well in image retrieval, the same rationale that object regions help avoid the influence

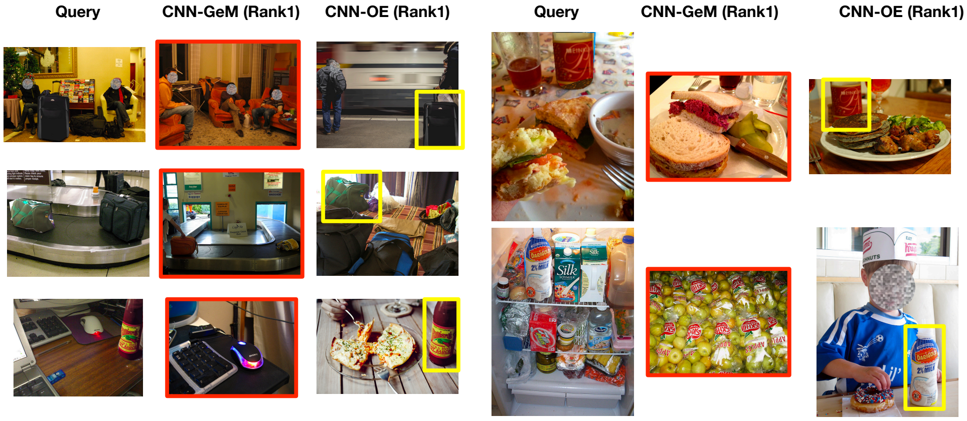


Figure 6.6: Example images and rank one results in the COCO-Fake dataset. First and the fourth column shows query tampered images, second and fifth column show rank-1 result from CNN-GeM; third and sixth shows rank-1 result from CNN-OE. The red border indicates incorrect matches and the yellow bounding box shows the matching objects. The faces are masked for privacy reason.

of background clutter should also apply. This capability has an important application in detecting tampered images that contain spliced objects [50], where a given image can be queried against a repository of images to detect near-duplicate object associations. Due to the proliferation of social media platforms, such application is becoming increasingly important, where it has been shown that there is a strong correlation between tampered images and the spread of misinformation.

To demonstrate the effectiveness of our approach for near-duplicate object retrieval, we conduct experiments on two different benchmarks. (1) **COCO-Fake**: we use the method described in [**?**] to generate 58 synthesized images with spliced objects as query images, and use images from COCO as database. We did not include any background images corresponding to the queries as our goal is to test on the ability to retrieve the donor images, from which the spliced objects originated. (2) Photoshop Image Retrieval dataset (**PIR**). The images are collected from the publicly available PS-Battles dataset [158] by selecting 3,278 original images as queries and 60,550 tampered images as the database. Each query has at least ten tampered versions in the database.

Table 6.7 shows retrieval results compared to different image retrieval methods. Our approach achieves better performance on both benchmarks because it can retrieve small spliced objects as a result of the hard attention provided by the detection model. Figure 6.6 shows some examples of the retrieval result. The first and the fourth columns are the query images; second and fifth columns show rank-1 retrieved results by CNN-MAC. CNN-MAC retrieves images with similar scenes but fails to retrieve tampered images that contain the spliced objects from the query image. The third and sixth columns show the rank-1 results retrieved by CNN-OE.

6.6 Conclusion

We provided analysis of embeddings learned from different models and demonstrated that embeddings learned from detection models are less discriminative than their classification counterparts. Based on our analysis, we proposed an approach that uses detection as a source of hard attention to improving retrieval performance. Our results showed that the proposed approach achieves state-of-the-art performance on different retrieval benchmarks. For applications with efficiency requirements, we have also introduced a student-teacher training regime that only needs a single forward pass during inference. Lastly, we show how our approach can be applied to near-duplicate object retrieval.

Chapter 7: Conclusion and Future Research Direction

In this dissertation, we mainly focus on image-geo localization and its application to media forensics. We first describe an application system that utilizes image geo-localization in Chapter 2. In Chapter 3 and Chapter 4, we then describe how such a system can be vulnerable under metadata tampering attack, and addressing the metadata tampering detection problem. In Chapter 5, we further develop an algorithm that can generate additional training data to improve the tampering detection algorithm. Finally, in Chapter 6, we describe an alternative approach, object provenance, that can be useful for search tampering images from a large-scale database. Image tampering detection is a challenging problem and we are far from solving it, especially when the tampering technique is rapidly advance along with detection algorithms. For example, recent advances in generative adversarial networks can generate realistic images with small computational effort compared to the traditional computer graphic approach. One particularly interesting approach describe in Chapter 6, provenance search, shows promising direction for tampering detection, however, there are many questions we also need to address when this approach is used for large-scale application, such as how are we generate a provenance database, how we do efficient search, and how to improve the recall and precision of the search. Finally, it worth note that another orthogonal direction is to generalize these approaches to video as video becomes more and more important in our daily digital life.

Bibliography

[1] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Learning Rich Features for Image Manipulation Detection. *arXiv:1805.04953 [cs]*, May 2018. URL `http://arxiv.org/abs/1805.04953`. arXiv: 1805.04953.

[2] Daniel Maier and Alexander Kleiner. Improved gps sensor model for mobile robots in urban terrain. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4385–4390. IEEE, 2010.

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

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

[5] Grant Schindler, Matthew Brown, and Richard Szeliski. City-scale location recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.

[6] Gerald Friedland, Oriol Vinyals, and Trevor Darrell. Multimodal location estimation. In *Proceedings of the international conference on Multimedia*, pages 1245–1252. ACM, 2010.

[7] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[8] Quan Fang, Jitao Sang, and Changsheng Xu. Giant: Geo-informative attributes for location recognition and exploration. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 13–22. ACM, 2013.

[9] Tsung-Yi Lin, Yin Cui, Serge Belongie, James Hays, and Cornell Tech. Learning deep representations for ground-to-aerial geolocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5007–5015, 2015.

[10] Yan-Tao Zheng, Ming Zhao, Yang Song, Helmut Adam, Ulrich Buddemeier, Alessandro Bissacco, Fernando Brucher, Tat-Seng Chua, and Hartmut Neven. Tour the world: building a web-scale landmark recognition engine. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1085–1092. IEEE, 2009.

[11] Yunpeng Li, David J Crandall, and Daniel P Huttenlocher. Landmark classification in large-scale image collections. In *Computer vision, 2009 IEEE 12th international conference on*, pages 1957–1964. IEEE, 2009.

[12] David M Chen, Georges Baatz, Kevin Köser, Sam S Tsai, Ramakrishna Vedantham, Timo Pylvä, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, et al. City-scale landmark identification on mobile devices. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 737–744. IEEE, 2011.

[13] Yan Chen, Jichang Zhao, Xia Hu, Xiaoming Zhang, Zhoujun Li, and Tat-Seng Chua. From interest to function: Location estimation in social media. In *AAAI*. Citeseer, 2013.

[14] Francine Chen, Dhiraj Joshi, Yasuhide Miura, and Tomoko Ohkuma. Social media-based profiling of business locations. In *Proceedings of the 3rd ACM Multimedia Workshop on Geotagging and Its Applications in Multimedia*, pages 1–6. ACM, 2014.

[15] Mans Hulden, Miikka Silfverberg, and Jerid Francom. Kernel density estimation for text-based geolocation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[16] Grant DeLozier, Jason Baldridge, and Loretta London. Gazetteer-independent toponym resolution using geographic word profiles. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[18] Li-Jia Li, Hao Su, Li Fei-Fei, and Eric P Xing. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *Advances in neural information processing systems*, pages 1378–1386, 2010.

[19] Jiawei Chen, Yin Cui, Guangnan Ye, Dong Liu, and Shih-Fu Chang. Event-driven semantic concept discovery by exploiting weakly tagged internet images. In *Proceedings of International Conference on Multimedia Retrieval*, page 1. ACM, 2014.

[20] Guangnan Ye, Yitong Li, Hongliang Xu, Dong Liu, and Shih-Fu Chang. Eventnet: A large scale structured concept library for complex event detection in video. *arXiv preprint arXiv:1506.02328*, 2015.

[21] Ali S Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[23] 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.

[24] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python.* " O'Reilly Media, Inc.", 2009.

[25] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. An empirical study of geographic user activity patterns in foursquare. *ICwSM*, 11:70–573, 2011.

[26] http://abcnews.go.com/wnt/story?id=1848351&page=1.

[27] Wei-Ta Chu, Xiang-You Zheng, and Ding-Shiuan Ding. Image2weather: A large-scale image dataset for weather property estimation. In *Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on*, pages 137–144. IEEE, 2016.

[28] Anna Volokitin, Radu Timofte, and Luc Van Gool. Deep features or not: Temperature and time prediction in outdoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 63–71, 2016.

[29] Cewu Lu, Di Lin, Jiaya Jia, and Chi-Keung Tang. Two-class weather classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3718–3725, 2014.

[30] Zheng Zhang and Huadong Ma. Multi-class weather classification on single images. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 4396–4400. IEEE, 2015.

[31] Nathan Jacobs, Nathaniel Roman, and Robert Pless. Consistent temporal variations in many outdoor scenes. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE, 2007.

[32] Weather underground: www.wunderground.com. URL `www.wunderground.com`.

[33] Husrev T Sencar and Nasir Memon. Overview of state-of-the-art in digital image forensics. *Algorithms, Architectures and Information Systems Security*, 3:325–348, 2008.

[34] Pravin Kakar and N Sudha. Verifying temporal data in geotagged images via sun azimuth estimation. *IEEE Transactions on Information Forensics and Security*, 7(3):1029–1039, 2012.

[35] Jean-François Lalonde, Alexei A Efros, and Srinivasa G Narasimhan. Estimating natural illumination from a single outdoor image. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 183–190. IEEE, 2009.

[36] Mohammad Islam, Nathan Jacobs, Hui Wu, and Richard Souvenir. Images+ weather: Collection, validation, and refinement. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop on Ground Truth*, volume 6, page 2, 2013.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[38] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[39] Daniel Glasner, Pascal Fua, Todd Zickler, and Lihi Zelnik-Manor. Hot or not: Exploring correlations between appearance and temperature. In *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015.

[40] Pysolar: http://pysolar.org/.

[41] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.

[42] Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Improved dct coefficient analysis for forgery localization in jpeg images. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 2444–2447. IEEE, 2011.

[43] Miroslav Goljan and Jessica Fridrich. Cfa-aware features for steganalysis of color images. In *Media Watermarking, Security, and Forensics 2015*, volume 9409, page 94090V. International Society for Optics and Photonics, 2015.

[44] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.

[45] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Splicebuster: A new blind image splicing detector. In *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on*, pages 1–6. IEEE, 2015.

[46] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, pages 159–164. ACM, 2017.

[47] Yuan Rao and Jiangqun Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In *Information Forensics and Security (WIFS), 2016 IEEE International Workshop on*, pages 1–6. IEEE, 2016.

[48] Ronald Salloum, Yuzhuo Ren, and C-C Jay Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). *Journal of Visual Communication and Image Representation*, 51:201–209, 2018.

[49] Jawadul H Bappy, Amit K Roy-Chowdhury, Jason Bunk, Lakshmanan Nataraj, and BS Manjunath. Exploiting spatial structure for localizing manipulated image regions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4970–4979, 2017.

[50] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. *arXiv preprint arXiv:1805.04953*, 2018.

[51] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[52] Aparna Bharati, Daniel Moreira, Joel Brogan, Patricia Hale, Kevin W Bowyer, Patrick J Flynn, Anderson Rocha, and Walter J Scheirer. Beyond pixels: Image provenance analysis leveraging metadata. *arXiv preprint arXiv:1807.03376*, 2018.

[53] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. Fighting fake news: Image splice detection via learned self-consistency. *arXiv preprint arXiv:1805.04096*, 2018.

[54] Xiaopeng Li, Wenyuan Xu, Song Wang, and Xianshan Qu. Are you lying: Validating the time-location of outdoor images. In *International Conference on Applied Cryptography and Network Security*, pages 103–123. Springer, 2017.

[55] Bor-Chun Chen, Pallabi Ghosh, Vlad I Morariu, and Larry S Davis. Detection of metadata tampering through discrepancy between image content and metadata using multi-task deep learning. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1872–1880. IEEE, 2017.

[56] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[57] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Largescale image retrieval with attentive deep local features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3456–3465, 2017.

[58] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*, pages 37–55. Springer, 2016.

[59] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[60] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 317–326, 2016.

[61] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.

[62] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1457, 2015.

[63] Xiyang Dai, Joe Yue-Hei Ng, and Larry S Davis. Fason: First and second order information fusion network for texture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7352–7360, 2017.

[64] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

[65] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European conference on computer vision*, pages 143–156. Springer, 2010.

[66] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *European conference on computer vision*, pages 589–600. Springer, 2006.

[67] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[68] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[69] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer, 2016.

[70] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 7, 2017.

[71] Peter J. Burt and Edward H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, October 1983. ISSN 07300301. doi: 10.1145/245.247. URL `http://portal.acm.org/citation.cfm?doid=245.247`.

[72] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on graphics (TOG)*, 22(3):313–318, 2003.

[73] Jean-Francois Lalonde and Alexei A. Efros. Using Color Compatibility for Assessing Image Realism. pages 1–8. IEEE, 2007. ISBN 978-1-4244-1630-1. doi: 10.1109/ICCV.2007.4409107. URL `http://ieeexplore.ieee.org/document/4409107/`.

[74] Micah K Johnson and Hanspeter Pster. CG2real: Improving the Realism of Computer Generated Images using a Large Collection of Photographs. page 13.

[75] Su Xue, Aseem Agarwala, Julie Dorsey, and Holly Rushmeier. Understanding and Improving the Realism of Image Composites. page 10.

[76] Jean-François Lalonde, Derek Hoiem, Alexei A Efros, Carsten Rother, John Winn, and Antonio Criminisi. Photo clip art. *ACM transactions on graphics (TOG)*, 26(3):3, 2007.

[77] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Learning a Discriminative Model for the Perception of Realism in Composite Images. *arXiv:1510.00477 [cs]*, October 2015. URL `http://arxiv.org/abs/1510.00477`. arXiv: 1510.00477.

[78] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep Image Harmonization. *arXiv:1703.00069 [cs]*, February 2017. URL `http://arxiv.org/abs/1703.00069`. arXiv: 1703.00069.

[79] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[80] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv:1611.07004 [cs]*, November 2016. URL `http://arxiv.org/abs/1611.07004`. arXiv: 1611.07004.

[81] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks. *arXiv:1612.05424 [cs]*, December 2016. URL `http://arxiv.org/abs/1612.05424`. arXiv: 1612.05424.

[82] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative Image Inpainting With Contextual Attention. page 10.

[83] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. *arXiv:1711.11585 [cs]*, November 2017. URL `http://arxiv.org/abs/1711.11585`. arXiv: 1711.11585.

[84] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv:1703.10593 [cs]*, March 2017. URL `http://arxiv.org/abs/1703.10593`. arXiv: 1703.10593.

[85] Qifeng Chen and Vladlen Koltun. Photographic Image Synthesis with Cascaded Refinement Networks. *arXiv:1707.09405 [cs]*, July 2017. URL `http://arxiv.org/abs/1707.09405`. arXiv: 1707.09405.

[86] Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric Image Synthesis. page 9.

[87] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. ST-GAN: Spatial Transformer Generative Adversarial Networks for Image Compositing. *arXiv:1803.01837 [cs]*, March 2018. URL `http://arxiv.org/abs/1803.01837`. arXiv: 1803.01837.

[88] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial Transformer Networks. *arXiv:1506.02025 [cs]*, June 2015. URL `http://arxiv.org/abs/1506.02025`. arXiv: 1506.02025.

[89] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. *ACM Transactions on Graphics (ToG)*, 23(3):294–302, 2004.

[90] Fuwen Tan, Crispin Bernier, Benjamin Cohen, Vicente Ordonez, and Connelly Barnes. Where and Who? Automatic Semantic-Aware Person Composition. page 10.

[91] Paul Debevec. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography. *conference proceedings*, page 10.

[92] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering Synthetic Objects into Legacy Photographs. page 12.

[93] Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig, and David Forsyth. Automatic Scene Inference for 3d Object Compositing. *ACM Transactions on Graphics*, 33(3): 1–15, June 2014. ISSN 07300301. doi: 10.1145/2602146. URL http://dl.acm.org/citation.cfm?doid=2631978.2602146.

[94] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-Francois Lalonde. Deep Outdoor Illumination Estimation. pages 2373–2382. IEEE, July 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.255. URL http://ieeexplore.ieee.org/document/8099738/.

[95] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Jonathan Eisenmann, Matt Fisher, Emiliano Gambaretto, Sunil Hadap, and Jean-François Lalonde. A Perceptual Measure for Deep Single Image Camera Calibration. *arXiv:1712.01259 [cs]*, December 2017. URL http://arxiv.org/abs/1712.01259. arXiv: 1712.01259.

[96] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[97] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context Encoders: Feature Learning by Inpainting. *arXiv:1604.07379 [cs]*, April 2016. URL http://arxiv.org/abs/1604.07379. arXiv: 1604.07379.

[98] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics*, 36(4):1–14, July 2017. ISSN 07300301. doi: 10.1145/3072959.3073659. URL http://dl.acm.org/citation.cfm?doid=3072959.3073659.

[99] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. pages 105–114. IEEE, July 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.19. URL http://ieeexplore.ieee.org/document/8099502/.

[100] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *arXiv:1603.08155 [cs]*, March 2016. URL `http://arxiv.org/abs/1603.08155`. arXiv: 1603.08155.

[101] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. SPECTRAL NORMALIZATION FOR GENERATIVE ADVERSARIAL NETWORKS. page 26, 2018.

[102] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[103] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[104] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018.

[105] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

[106] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[107] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[108] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

[109] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. Factors of transferability for a generic convnet representation. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1790–1802, 2016.

[110] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? *arXiv preprint arXiv:1805.08974*, 2018.

[111] Yong Rui, Thomas S Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*, 10(1):39–62, 1999.

[112] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003.

[113] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Improving bag-of-features for large scale image search. *International journal of computer vision*, 87(3): 316–336, 2010.

[114] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2011.

[115] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer, 2014.

[116] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 1269–1277, 2015.

[117] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015.

[118] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[119] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *European conference on computer vision*, pages 241–257. Springer, 2016.

[120] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European conference on computer vision*, pages 3–20. Springer, 2016.

[121] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3456–3465, 2017.

[122] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837, 2017.

[123] Xu Zhang, Felix X Yu, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4595–4603, 2017.

[124] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 360–368, 2017.

[125] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017.

[126] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2593–2601, 2017.

[127] Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–285, 2018.

[128] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6148–6157, 2017.

[129] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6917–6925, 2018.

[130] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. Dgc-net: Dense geometric correspondence network. *arXiv preprint arXiv:1810.08393*, 2018.

[131] Liang Zheng, Yi Yang, and Qi Tian. Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1224–1244, 2018.

[132] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[133] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *international Conference on computer vision & Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE Computer Society, 2005.

[134] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models.

IEEE transactions on pattern analysis and machine intelligence, 32(9):1627–1645, 2009.

[135] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1532–1545, 2014.

[136] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[137] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[138] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In *Advances in Neural Information Processing Systems*, pages 9333–9343, 2018.

[139] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165*, 2018.

[140] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. *arXiv preprint arXiv:1812.01584*, 2018.

[141] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

[142] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. ACM, 2006.

[143] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014.

[144] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[145] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[146] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems*, pages 742–751, 2017.

[147] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[148] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[149] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[150] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5706–5715, 2018.

[151] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4004–4012, 2016.

[152] Hervé Jégou and Ondřej Chum. Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening. In *European conference on computer vision*, pages 774–787. Springer, 2012.

[153] Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4(3):251–258, 2016.

[154] Filip Radenović, Giorgos Tolias, and Ondrej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[155] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[156] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[157] Xun Wang, Xintong Han, Weiling Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. *arXiv preprint arXiv:1904.06627*, 2019.

[158] Silvan Heller, Luca Rossetto, and Heiko Schuldt. The PS-Battles Dataset – an Image Collection for Image Manipulation Detection. *CoRR*, abs/1804.04866, 2018. URL `http://arxiv.org/abs/1804.04866`.