VISUAL ATTRIBUTE DISCOVERY AND ANALYSES FROM WEB-DATA

Sirion Vittayakorn

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2016

Approved by:

Tamara L. Berg

Alexander C. Berg

Jan-Michael Frahm

James Hays

Sanja Fidler

ABSTRACT

SIRION VITTAYAKORN: VISUAL ATTRIBUTE DISCOVERY AND ANALYSES FROM WEB-DATA.
(Under the direction of Tamara L. Berg.)

Visual attributes are important for describing and understanding an objects appearance. For an object classification or recognition task, an algorithm needs to infer the visual attributes of an object to compare, categorize or recognize the objects. In a zero-shot learning scenario, the algorithm depends on the visual attributes to describe an unknown object since the training samples are not available. Because different object categories usually share some common attributes (e.g., many animals have four legs, a tail and fur), the act of explicitly modeling attributes not only allows previously learnt attributes to be transferred to a novel category but also reduces the number of training samples for the new category which can be important when the number of training samples is limited. Even though larger numbers of visual attributes help the algorithm to better describe an image, they also require a larger set of training data. In the supervised scenario, data collection can be both a costly and time-consuming process. To mitigate the data collection costs, this dissertation exploits the weakly-supervised data from the Web in order to construct computational methodologies for the discovery of visual attributes, as well as an analysis across time and domains.

This dissertation first presents an automatic approach to learning hundreds of visual attributes from the open-world vocabulary on the Web using a convolutional neural network. The proposed method tries to understand visual attributes in terms of perception inside deep neural networks. By focusing on the analysis of neural activations, the system can identify the degree to which an attribute can be visually perceptible and can localize the visual attributes in an image. Moreover, the approach exploits the layered structure of the deep model to determine the semantic depth of the attributes.

Beyond visual attribute discovery, this dissertation explores how visual styles (i.e., attributes that correspond to multiple visual concepts) change across time. These are referred to as visual trends. To this goal, this dissertation introduces several deep neural networks for estimating when objects were made together with the analyses of the neural activations and their degree of entropy to gain insights into the deep network. To utilize the dating of the historical object frameworks in real-world applications, the dating frameworks are applied to analyze the influence of vintage fashion on runway collections, as well as to analyze the influence of fashion on runway collections and on street fashion.

Finally, this dissertation introduces an approach to recognizing and transferring visual attributes across domains in a realistic manner. Given two input images from two different domains: 1) a shopping image, and 2) a scene image, this dissertation proposes a generative adversarial network for transferring the product pixels from the shopping image to the scene image such that: 1) the output image looks realistic and 2) the visual attributes of the product are preserved.

In summary, this dissertation utilizes the weakly-supervised data from the Web for the purposes of visual attribute discovery and an analysis across time and domains. Beyond the novel computational methodology for each problem, this dissertation demonstrates

that the proposed approaches can be applied to many real-world applications such as dating historical objects, visual trend prediction and analysis, cross-domain image label transfer, cross-domain pixel transfer for home decoration, among others.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my adviser, Professor Tamara Berg. Tamara has been supportive of me even before my official acceptance letter had arrived. She helped me go through the difficult and stressful admission process while I was situated half-way around the world. During my graduate years, Tamara has always provided me with inspiration and encouragement that have helped me to explore novel directions in my research. Both her expertise and eagerness have enlightened my research in many ways. Her high-level guidance and insightful ideas have always helped me to move forward in a promising direction. I sincerely appreciate all her patience, generosity, and above all, the professional and personal support that she extended to me every single day of my graduate-level education. I would like to thank my co-adviser Professor Alex Berg whose knowledge and expertise has significantly contributed to the success of this dissertation. I am so grateful for his insights, comments and supports throughout these years.

My sincere gratitude also goes to the dissertation committee members: Professor Jan-Michael Frahm, Professor James Hays, and Professor Sanja Fidler for their enthusiasm toward my research and for their insightful comments, which both challenged and enriched my own understanding of this field of study and my research. Special thanks are extended to Professor James Hays who has not only been a key member of my disserta-

tion committee, but has also been my adviser during the pursuit of my Masters degree at Brown. The computer vision course I studied with James introduced me to the idea of just how fascinating the concept of computer vision can be. His work inspired me to exploit large-scale data from the Web. His advice and recommendations have helped me in my pursuit of my Ph.D program. This was done along with my wonderful adviser, Professor Tamara Berg. I am grateful to have had James of this committee and as an adviser. I would also like to especially thank Professor Sanja Fidler, who further contributed to my research by engaging in hours of invaluable collaboration and by offering constructive feedback. I am thankful for her insights, brilliant project ideas and suggestions. Special thanks to Professor Sanpawat Kantabutra for his support and providing me with the opportunity to conduct research during my undergraduate years.

I want to thank Kota Yamaguchi for being a great friend and an excellent mentor for me. Kota taught me how to collect an immeasurable amount of data from the web, how to write better code and helped me with all the technical difficulties that came up during my first year at Stony Brook and during my internship in Japan. I truly appreciate and value all your patience and help.

I would like to thank all my fellow students both at Stony Brook and Chapel Hill. Thanks to Vicente Ordonez and Xufeng Han who gave me advice, comments and insights into my research and about life in general. Thanks to my roommates and colleagues, Hadi Kiapour and Wei Liu, for their professional and personal support. Thanks to Yipin Zhou for giving me advice on my research, for our trips together to various restaurants and the countless hours that were spent driving me home during the many late nights I encountered during my studies. It was pleasant to spend time in the wonderful company of many individuals including Licheng Yu, Eunbyung Park, Patrick Poirson, Philip Ammirato, Cheng-Yang Fu, Tatiana Tommasi, Hyo Jin Kim, Dinghuang Ji and many others. Thank you for making my time during graduate school a truly memorable experience. I will always appreciate our time together and I hope that our friendships last a lifetime.

This long journey would not have been possible without the support from Thai government scholarship, faculty of information technology at King Mongkuts institute of Technology Ladkrabang, Somthavil Thongprasert and Supranee O'dell from office of educational affairs at royal Thai embassy.

Thanks to all my friends: Siripat, Sukree, Sutiney, Suthira, Supawan, Sangtip, Sopida and Arpaporn for their lifelong friendship and supports since the day we met at Satit CMU. Also thanks to Janrapee, Narumol and Tussanawalai that always made our winter trips to the beaches memorable and fun.

Reflecting on my long education path, I want to pay a special thanks to Bundit for everything always. His support, understanding and patience always kept me moving forward toward my goal. Beside my family, he is the most important source of strength through my ups and downs. Thank you for always being there for me. Finally, and most importantly I would like to express my deepest gratitude to my family: my big brother, Naratip, for his advice and support about this research pursuit and about life in general and my parents, Niphon and Soisuda, for always believing in me and supporting me at every step of my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Visual attributes are the attributes that human use to describe the visual appearances of objects. For many computer vision tasks, such as object classification or recognition, the algorithm needs to infer what the visual attributes of the object are (e.g., a banana is yellow, elongated and curved, while a strawberry is conical, bright red with its seeds on the outer skin) in order to compare, categorize or recognize the objects. In a zero-shot learning scenario where the system faces an unknown category with zero training data, visual attributes are crucial for describing and understanding the object appearance even though the system cannot identify it. For example, we can describe the object in Figure 1.1 as an object with two black eyes, two ears, four papaya whip small horns, periwinkle fluffy fur and a rounded tail. Moreover, since different object categories usually share some common attributes (e.g., many animals have four legs, a tail and fur), explicitly modeling attributes not only allows previously learnt attributes to be transferred to a novel category but also reduces the number of training samples for the new category which can be important when the number of training samples are limited.

Although the visual attributes are important in many tasks, visual attributes discovery is a challenging task. While one can describe handbag in Figure 1.2 as 'the smoky gray fades into a pearly white, resembling the majestic, snow-capped Himalayas, another one can describe the same handbag as 'a 25 cm handbag of white matte niloticus

Figure 1.1: Unknown object.

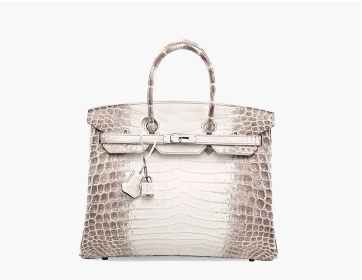
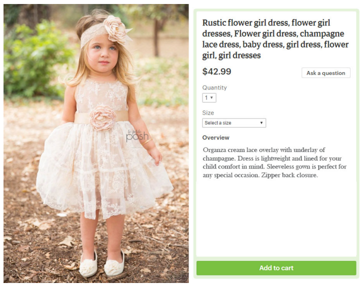


Figure 1.2: Birkin bag.



Rustic flower girl dress, flower girl dresses, Flower girl dress, champagne lace dress, baby dress, girl dress, flower girl, girl dresses

$42.99

Quantity
1

Size
Select a size

Overview

Organza cream lace overlay with underlay of champagne. Dress is lightweight and lined for your child comfort in mind. Sleeveless gown is perfect for any special occasion. Zipper back closure.

Figure 1.3: Example of weakly-supervised data.

crocodile skin with palladium hardware as well as other hundreds different ways. It is obvious to say that the more visual attributes the system can learn, the better the system can describe an object. However, if an object can be described by tens or hundreds of attributes, the number of attributes that use to describe tens or hundreds of objects must be very large. And it is very expensive to learn such large number of visual attributes from supervised data. What is an alternative data source for visual attributes discovery?

Nowadays, we are living in the time that millions of images are uploaded to the Internet every day. The online photo sharing websites (e.g., Flickr, Pinterest, etc.), the social media networks (e.g., Facebook, Instagram, Chictopia, etc.), the online collections from museum (e.g., MET, Europeana, etc.), the online magazines (e.g., Vogue, Cosmopolitan, etc.) or e-commerce websites (e.g., etsy, houzz, etc.) have millions of images with millions of active users. In 2016, Facebook users have shared more than 240,000 new photos per day, resulting in about 250 billion photos one this website alone. Similarly to Facebook, the online mobile photo sharing application Instagram has also become popular in the past couple of years. Instagram users upload more than 80 million photos daily, which amounts to 40 billion shared photos in total. With a terabyte of photo storage and more flexible photo and video resolution options than Facebook and Instagram, Flickr now hosts more than 12 billion photos with 2 million new photos posted every day. Unlike other photo sharing websites, Pinterest users are creating more than 5 million photo catalogs daily.

Not only have billions of photos been uploaded to the Internet, text descriptions related to these photos are uploaded as well. When people share images and details of their lives or their interests via social networks, it is not just photos but also the story behind those photos. A short caption or tag is provided alongside photos, which can describe the content of the photo or the story related to the photo. These photos and textual descriptions make Web data a rich source of weakly-supervised data as shown in Figure 1.3. The characteristics of the internet photos and their associated text are appealing to researchers for many reasons:

Data size There are already billions of photos and their associated texts available on the Internet and this number is increasing every second. With a tremendous amount of data, a variety of data-driven approaches have become feasible.

Data availability Since people use social networks to share their personal stories or interests with others, these photos are mostly publicly available and easy to collect.

Cost efficiency Collecting photos and texts associated with them from the web is more affordable than manual annotation (supervised data). Although web data can be noisy and some descriptions or tags might not be related to the image content, verifying image description is cheaper than writing the image description task.

Data diversity With billions of users worldwide, the photos and their descriptions that are posted are diverse. These photos cover thousands of topics (e.g., activities or events, landscapes or landmarks, animals, objects, people, etc.), involve hundreds of photograph styles (e.g., black/white, macro, HDR, time lapse, motion, panorama, portrait, etc.) and different timestamps (from the past to the present) from all over the world.

The objective of this dissertation is to exploit the weakly-supervised data from the Web in order to construct computational methodologies so as to: 1) discover and recognize visual attributes of objects (Chapter 3), 2) recognize and analyses the visual styles (i.e., attributes that corresponses to multiple visual attributes) across time or visual trends (Chapter 4), and 3) synthesize the images of objects across image domain while preserving their visual attributes (Chapter 5).

1.1 Thesis statement

By exploiting a large-scale weakly-supervised data from the Web using the deep convolutional network, it is feasible to develop the automatic systems to discover hundreds of visual attributes together with recognize and analyses the visual attributes across time and image domain.

1.2 Outline of contributions

This dissertation begins by reviewing the relevant work discussed in Chapter 2. The chapter will first go through the recent work on the visual attributes learning and the convolutional neural networks for visual attribute recognition. Since the visual attributes of objects are contextually dependent on object domain, this dissertation studies the

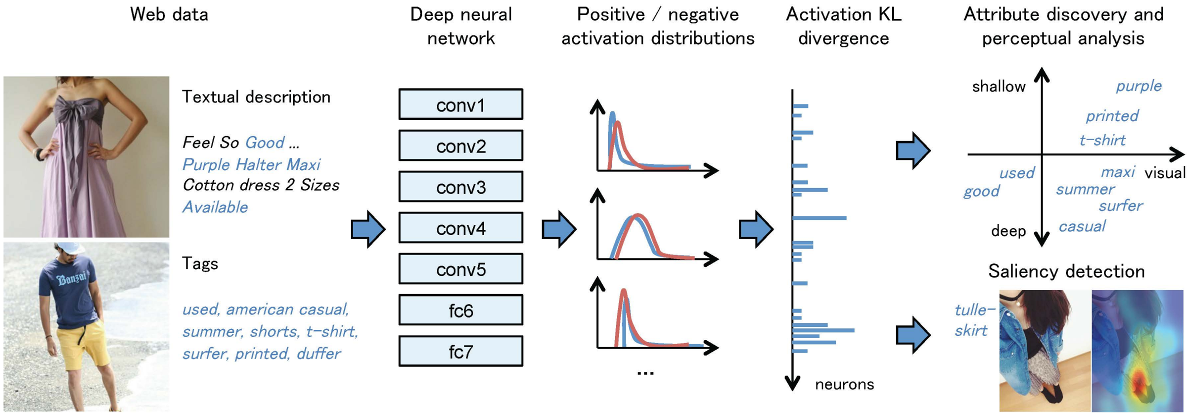


Figure 1.4: Visual attribute discovery framework.

consistent meaning of the attributes under the clothing domain to avoid a semantic shift. Thus, Chapter 2 reviews several methods used for clothing recognition and its applications. The chapter also includes state-of-the-art work on generative network for synthesizing images for different applications.

Then, the main contributions of this dissertation that advanced the state of the art in visual attribute discovery and analyses are introduced throughout the following chapters.

First, Chapter 3 describes an automatic approach to learning visual attributes from the open-world vocabulary on the Web. Although there have been numerous attempts at learning novel concepts from the Web in the past, this dissertation tries to better understand potentially-attributable words in terms of perception inside deep neural networks. Since deep networks have demonstrated outstanding performance in many computer vision tasks, this chapter focuses on the analysis of neural activations in order to identify the degree of being visually perceptible, namely the *visualness* of a given attribute. The proposed approach takes advantage of the layered structure of the deep model to determine the semantic depth of the attribute.

The experiments show that by using a trained neural network, a visual attribute word can be characterized using the divergence of neural activations in the weakly-annotated data. Figure 1.4 illustrates the visual attribute discovery framework. The approach starts by cleaning the noisy Web data to identify and select the potentially visual attributes in the dataset. Then, it splits the data into positive and negative sets. Using a pre-trained neural network, highly-activating neurons are identified by the KL divergence of the activations. The results show that the identified neurons (*prime units*) can be used for: 1) learning a novel attribute classifier that is close to human perception, 2) understanding the perceptual depth of the attribute, and 3) identifying attribute-specific saliency in the image.

Although the results from Chapter 3 show that visual attributes can be automatically discovered from Web data using the neural activation acquired from a deep network, the discovered visual attributes usually correspond to one visual representation (e.g., striped corresponds to long narrow bands of different colors). However, this characteristic does not stay true for attributes in every domain. Some attributes are more informative in that they correspond to multiple visual representations or concepts, for example, the temporal attributes. In some domains like fashion, the *1970s* corresponds to *mini-skirt, the disco-look, bell-bottoms, tight on top and loose on bottom, etc.*. These temporal attributes are very important. With the temporal attributes, it is feasible for researcher to explore several interesting problems such as dating historical photograph, temporal classification for data organization. Notably, temporal attributes become important in the fashion domain where the reoccurrence of visual concepts from time to time has been observed.

The temporal attribute also leads to appealing to certain applications, such as in trend analysis and prediction problems.

In order to conduct the temporal analyses of the visual concepts, Chapter 4 first explores the deep learning methods for the temporal estimation of clothing items. Then, the analyses of neural activations and their entropy from the temporal estimation network are provided in order to give an additional understanding of the network. To demonstrate the advantages of the temporal estimation network in real world applications, the analysis of the influence of vintage fashion in the fashion show collections is explored. The results show that the proposed approach can discover the degree of vintage influences that agrees with the reference.

To further extend the analysis of fashion in the real world, this chapter introduces an approach to the study of fashion both on the runway and in street settings. The contributions involve collecting a new fashion show dataset, designing features suitable for capturing outfit appearance, collecting human judgments of outfit similarity, and learning similarity functions on the features to mimic human judgments. The intrinsic evaluations of the learned models are provided to assess performance on outfit similarity prediction. Finally, an application that tracks visual trends as runway fashions filtering down to street fashions is described in this chapter.

Chapter 5 goes beyond the concept of learning visual attribute by presenting a generative model that is trained to transfer the visual attributes of the input object to a new image. Given two inputs: room image and object image (e.g., chandelier, nightstand, chair, etc.), the convolutional neural network is trained to generate an output image where the input object appears in the input room. The objective of the network is to generate the output image such that: 1) the output room looks realistic and 2) the visual attributes of the input objects are transferred into the synthesized object in the output image.

Finally, Chapter 6 will conclude this dissertation.

CHAPTER 2: BACKGROUND

This chapter will review the relevant work as it relates to three different topics: visual attribute learning, convolutional neural network and computer vision within the fashion domain.

2.1 Visual attribute learning

Visual attribute is an important cue involved in many tasks, such as object classification and recognition, fine-grained classification (Berg et al., 2010; Duan et al., 2012; Rastegari et al., 2012), face or person verification (Kumar et al., 2009, 2011; Taigman et al., 2014) and recognition (Liu et al., 2015; Taigman et al., 2014) or activity classification (Raptis et al., 2012), as well as a number of others. All of these tasks require informative visual attributes to distinguish target samples from other samples. The importance of visual attributes has encouraged a large number of research studies that have been used to generate a benchmark of attribute datasets related to visual attribute recognition and visual attribute discovery.

2.1.1 Visual attribute datasets

To evaluate the performance of visual attribute recognition and discovery across different approaches, several studies have attempted to propose a visual attribute dataset as a benchmark for the community.

The early work (Farhadi et al., 2009) has proposed two attribute datasets: 1) a-Pascal containing images of 20 classes from the PASCAL VOC2008 dataset (Everingham et al., 2008), and 2) a-Yahoo containing images of 12 additional classes acquired from Yahoo!. Both datasets have been manually annotated with 64 types of binary attributes. Several studies have provided attribute datasets in specific domains; 1) Animals (Lampert et al., 2009), 2) Human faces (Liu et al., 2015), 3) Clothing (Chen et al., 2012) and 4) Scene (Patterson and Hays, 2012, 2016). Animals with Attributes (Lampert et al., 2009) contains more than 30K images of 50 different animal classes acquired from the Internet and that have been manually labeled with 85 attributes. For the human faces domain, the CelebFaces Attributes Dataset (Liu et al., 2015) is a large-scale face attribute dataset with more than 200K celebrity images, each with 40 attribute annotations. In the clothing domain, the Clothing Attribute Dataset (Liu et al., 2015) is comprised of about 1,800 images of street fashion with 26 clothing attributes (e.g., 'Necktie', 'Collar' and 'Spotted Pattern'). To tackle the scene understanding problem and that of fine-grained scene recognition, the SUN Attribute database (Patterson and Hays, 2012) was the first large-scale scene attribute database on top of the fine-grained SUN categorical database (Xiao et al., 2014). The later work, the COCO attribute dataset (Patterson and Hays, 2016) was an attempt to discover and annotate visual attributes for the COCO dataset (Lin et al., 2014) with 3.5 million object-attribute pair annotations describing 180 thousand different objects. Recently, the Visual Genome dataset (Krishna et al., 2016) was presented to handle the cognitive tasks (e.g., image description and question answering). The dataset contains over 100K images with the dense annotations of objects, attributes, and pairwise relationships between objects.

2.1.2 Visual attribute recognition

Since visual attributes have been shown to be crucial for several tasks such as zero-shot learning when training samples are not available or fine-grain classification when only class label is not enough, the automatic identification of such attributes has been the focus of several research studies.

Earlier studies (Farhadi et al., 2009; Lampert et al., 2009) go beyond naming the objects to describe the objects by using their visual attributes and train the models for the unseen object categories based on the visual attributes. Zero-shot learning or object recognition with no exist training examples has been explored in many later studies (Lampert et al., 2014; Socher et al., 2013; Yu et al., 2013). For example, the attribute classifiers which are trained on a high-level description that is phrased in terms of the semantic attributes (Lampert et al., 2014). The attribute classifiers can be trained independently from the existing image data sets and new classes can be detected based on their attribute representation. More recent works apply convolutional neural networks to tackle the attribute recognition problem to both weakly-supervised (Shankar et al., 2015) and unsupervised scenario (Huang et al., 2016; Doersch et al., 2015). As opposed to predicting the presence or absence of visual attributes, many studies (Parikh and Grauman, 2011; Shrivastava et al., 2012) have explored the use of *relative attributes* which indicates the strength of an attribute in an image with respect to other images. Visual attributes can be used as a feedback for an image search where a user indicates which attributes of exemplar images should be tuned in order to improve the retrieval results (Kovashka et al., 2012).

2.1.3 Visual attribute discovery

Visual attributes are used to describe the visual appearances of an object. Some attributes are specific for certain object categories, while others are shared among categories. In order to classify or recognize an object category, the discriminative attributes are required. However, these informative attributes are category dependent. The discriminative attributes that can distinguish between oranges vs. apples can be different from those used to distinguish between oranges vs. watermelons. How one discovers the informative attributes for each task is still a challenging problem. Several approaches have been proposed to tackle this problem.

Text-based search retrieval

There have been several attempts at attribute discovery from a collection of images from the Web (Chen et al., 2013; Ferrari and Zisserman, 2007) based on the text-based search image retrieval. Ferrari *et al.* (2007) propose a probabilistic generative model of a given attribute (e.g., red, spotted), and a procedure for learning its parameters from images collected by a text-based retrieval. NEIL (Chen et al., 2013) aimed to discover common sense knowledge from the Web starting from small exemplar images per concept to train the initial detector. Through the use of initial detectors, the system automatically extracts common sense relationships between the object and the attributes. Both the detectors and the common sense relationships are then used to retrieve new images which in turn are used to re-train the detectors and so on.

Text data mining

Alternative approaches start from mining the attributes from the textual data (e.g., descriptions, labels or tags) and then training the visual classifier of each concept. LEVAN (Divvala et al., 2014) began by mining the bi-gram concepts from a large text corpus, and the automatically retrieving the training images from the Web to learn a full-fledged detection model for each concept. ConceptLearner (Zhou et al., 2015) uses weakly labeled image collection from Flickr to train visual concept detectors. Sun *et al.* (2015) take advantage of natural language by embedding the semantic similarity of the attributes into their pipeline. In the e-commerce scenario, Berg *et al.* (2010) presented an automatic system to identify visual attributes from noisy textual data and define the *visualness* of an attribute as an average value of the precision of the visual classifier on the held-out data.

Visual data mining

To bypass the noisy textual data, several visual data mining approaches have been explored to directly discover the visual elements which are the characteristics of a given category. To automatically locate the visual elements that are most geo-informative, Doersch *et al.* (2012) searched for the visual elements that are both: 1) *repeating*, i.e. they frequently occur in some geographic regions R, and 2) *geographically discriminative*, i.e. they occur much more often in R than in R^C. Given the randomly sampled candidate patches, a linear SVM detector is trained in a iterative manner to construct a geo-informative cluster. In contrast to Doersch *et al.* (2012) who searched for patterns that remain visually consistent throughout the dataset, Lee *et al.* (2013) targeted the visual elements whose appearance gradually changes through time and space, which are called *style-sensitive* elements. The system starts from mining the style-sensitive patches and incrementally builds correspondences between these patches to find the same elements across the dataset. The style-aware regressors are then trained to model each elements range of stylistic differences.

Although these discriminative elements are crucial and machine-detectable, they might not be human-understandable which limits the ability of humans to understand object models or contribute the domain knowledge to the recognition systems. Thus, Duan *et al.* (2012) have proposed an interactive system that discovers the discriminative attributes, which are both machine-detectable and human-understandable. At each iteration, the system starts by discovering the candidates' local attributes. These candidates are then presented to the human in order to collect attribute names. The candidates for which the users can give a name are added to the pool of attributes, while the unnamed ones are discarded. Unlike the other studies, Rastegari *et al.* (2012) discovered the visual attributes in the form of contrasts from the learned binary codes. Each bit in the binary code can be visualized as a hyper-plane in the feature space that corresponds to a visual attribute. The images from different sides of the hyper-plan correspond to different visual attributes, for example, images of *silver or metallic* objects vs. *natural* images.

2.2 Convolutional neural networks

Convolutional neural networks (CNNs) have a great impact on the computer vision community due to the outstanding performance of detection (Szegedy et al., 2015), semantic segmentation (Long et al., 2015) and classification (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; He et al., 2015) on the benchmark dataset that have been employed in the past couple of years.

The feature representations from the network trained on 1.2 million supervised images (Deng et al., 2009) have been shown to generalize well to other image classification tasks (Donahue et al., 2013), as well as to other related tasks such as object detection (Girshick et al., 2014; Sermanet et al., 2013), pose estimation and action detection (Gkioxari et al., 2014), or fine-grained category detection (Zhang et al., 2014). Karayev *et al.* (2014) stated that using a pre-trained network (Donahue et al., 2013) as a generic feature extractor, produces a better classifier for photo and painting style than hand-crafted features.

2.2.1 Understanding neural representation

Unlike certain hand-crafted representations such as SIFT (Lowe, 2004) or HOG (Dalal and Triggs, 2005), the learned representations from the deep networks are not immediately interpretable. Thus, the deep analyses of these intermediate representations of the neural networks have been explored in many recent research studies (Yosinski et al., 2015;

Zeiler and Fergus, 2014; Zhou et al., 2014).

To better understand what the network has learned, Fischer *et al.* (2014) compared the learned representation with SIFT in a descriptor matching task. Zeiler and Fergus (Zeiler and Fergus, 2014) proposed a novel technique to map the neural activation back to the image space while Yosinki *et al.* (2015) introduced the tools to visualize the neuron activations in real-time. Long *et al.* (2014) studied the effectiveness of the neural activation features for tasks requiring correspondence. Zhou *et al.* (2014) proposed a data-driven approach to visualize the receptive fields (RFs) of each neuron in the network and to finally find the object detectors that emerge in a scene classification network. Simonyan *et al.* (2014) visualized parts of the image that cause the highest change in the class labels that had been computed by back-propagation and applied this technique to compute an image-specific class saliency map which highlights the areas of the given image and discriminates with respect to the given class. Mahendran and Vedaldi (Mahendran and Vedaldi, 2015) extended this approach by introducing natural image priors which result in inverse images that have fewer artifacts. Dosovitskiy and Brox (2016b) proposed a deconvolutional network that could invert a CNN in a feed-forward manner.

2.2.2 Convolutional neural networks for attribute recognition

Recently, a number of research studies have attempted to tackle the attribute recognition problem using the convolutional neural network. Both Escorcia *et al.* (2015) and Ozeki *et al.* (2014) explored the relationship between neural representation and the visual attributes acquired from the supervised dataset. Escorcia *et al.* (2015) revealed empirical evidence on the existence of Attribute Centric Nodes (ACNs) within an object classification network. ACNs encode information that precisely reconstruct attributes in a sparse and unevenly distributed manner among the network layers. Ozeki *et al.* (2014) conducted several experiments in an attempt to demonstrate that the neural representations can be interpreted as category-level attributes which are informative for the classification task.

Shankar *et al.* (2015) proposed a deep network training procedure for the purpose of multiple attribute predictions acquired from weakly-supervised data. During training, the responses of the neuron activations are exploited to provide multiple pseudo-labels for training images in subsequent iterations. Doersch *et al.* (2015) explored the use of spatial context as a cue to train a convolutional neural network that can predict the relative position of the input patches. Although the network is trained for different tasks, the visual representation has also proven to be useful for both object detection and unsupervised object discovery. Huang *et al.* (2016) have proposed a two-stage pipeline, which consists of unsupervised discriminative clustering and weakly-supervised hashing, where the visual clusters, hashing functions and feature representations are jointly learned in order to learn the visual attributes from the unlabeled data. More recent works rely on convolutional neural network to tackle the attribute recognition problem on both weakly-supervised (Shankar et al., 2015) and unsupervised scenario (Huang et al., 2016).

2.2.3 Generative model

Due to a promising performance on generating natural images using a convolutional neural network, the generative models have recently gained attention from the computer vision community. Goodfellow *et al.* (2014) proposed a new framework for estimating generative models via an adversarial process (GAN) that jointly train two different models: a generative model G which captures the data distribution, and a discriminative model D which estimates the probability that a sample came from the training data rather than the G model. Inspired by GAN, Radford *et al.* (2016) proposed new convolutional architectures and optimization hyperparameters for GAN to improve the results. Denton *et al.* (2015) combined the GAN structure with the multi-scale Laplacian pyramid to produce high-resolution results. Dosovitskiy and Brox (2016a) introduced a new loss function called deep perceptual similarity metrics (DeePSiM) which compute the distances between image feature extracted by the encoder of the GAN-style network. The experiments showed that DeePSiM better reflects the perceptual similarity of the images and thus leads to better results. Nguyen *et al.* (2016) applied the network developed by Dosovitskiy and Brox (2016a) to explore the preferred input images of the classification networks e.g., CaffeNet (Jia et al., 2014).

Applying recurrent neural networks for image generation, Gregor *et al.* (2015) combined the sequential variational auto-encoding framework with an attention mechanism to iteratively construct parts of an image. Unlike some previous research studies, Dosovitskiy *et al.* (2015) demonstrated that the convolutional neural network can be trained to generate images of objects, given object types, viewpoints, and colors using supervised data.

2.3 Computer vision in fashion domain

One important characteristic of the visual concept is contextual dependency; the same concept can correspond to different visual elements depending on the context. For example, the term *red eye* can refer to an overnight airline flight or an eye that appears red due to illness or injury. This contextual dependency can result in the certain amount of ambiguity for the visual classifier (*red* classifier). To isolate the contextual dependency of the attributes to the object category, this dissertation focuses on domain-specific data, such as that of the fashion domain, which has gained an increasing level of interest from computer vision researchers, possibly because of the potential benefit in e-commerce applications. Thus, this section will review recent efforts in clothing recognition, clothing retrieval and their related applications.

2.3.1 Clothing recognition

Clothing recognition is one of the fundamental problems of computer vision research in the fashion domain. Several research studies have depended on clothing cues to: 1) Recognize the occupations of people (Song et al., 2011; Shao et al., 2013) or their social identity (Murillo et al., 2012; Kwak et al., 2013; Kiapour et al., 2014), 2) Recommend the fashion coordination by occasion (Liu et al., 2012a), or 3) Predict the fashionability of the users' photographs and to suggest subtle improvements that users could make to improve their appeal (Simo-Serra et al., 2015).

Many research studies have attempted to tackle the clothing parsing problem through the use of large clothing categories (Yamaguchi et al., 2012, 2013; Simo-Serra et al., 2014). The earlier research study by (Yamaguchi et al., 2012) formulated a way to addressing the problem as a MAP estimation of image-region labels in the conditional random field (CRF) given pose estimations. Dong *et al.* (2013) proposed clothing parsing as an inference problem over *parselets*, which is the basis group of image regions that constitute clothing items. Also, Liu *et al.* (2014) presented a pipeline in order to eliminate a pixel-level supervision in learning how to use image-level color tags. The later work from (Yamaguchi et al., 2013) tackled the problem of using a retrieval based approach that combines clothing parsing from: pre-trained global clothing models, local clothing models learned on the fly from the retrieved examples, and transferred parse masks from retrieved examples. Simo-Serra *et al.* (2014) presented the parsing problem as one of inference in a pose-aware CRF which exploits appearance, figure/ground segmentation, shape and location priors for each garment as well as the similarities between segments, and symmetries between different human body parts.

2.3.2 Clothing retrieval

In a similar fashion to the clothing recognition task, clothing retrieval has also become more popular due to the growth of e-commerce and mobile applications. This has resulted in an increasing number of recent studies on clothing retrieval and its application.

The street-to-shop clothing retrieval system developed by Liu *et al.* (2012b) aims to connect street-fashion snapshots with the images that have been acquired from the online shopping website using hand-crafted features on the supervised dataset. They employ a mapping function that involves street and shopping images using a sparsely coded transfer matrix to mitigate the cross-domain effect on the retrieval results. Kalantidis *et al.* (2013) also proposed a cross-domain retrieval system of street fashion and shopping images and suggested visually similar outfits from the shop to the user. Cushen *et al.* (2013) proposed a visual search approach with efficiency in a mobile scenario. Kiapour *et al.* (2015) approached the problem using a deep network to learn the similarity measurement between the two image domains, comprised of the street and shop images.

CHAPTER 3: AUTOMATIC ATTRIBUTE DISCOVERY WITH NEURAL ACTIVATIONS

This chapter introduces an automatic approach to learn visual attribute words from the open-world vocabulary available on the Web. There have been numerous attempts at learning novel concepts from the Web in the past (Chen et al., 2013; Ferrari and Zisserman, 2007; Divvala et al., 2014; Zhou et al., 2015; Berg et al., 2010). However, none of the previous works have aimed to understand potential attribute words in terms of the perception found inside deep neural networks, which demonstrate outstanding performance in object recognition (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; He et al., 2015) from supervised data, and from noisy data (Tong Xiao et al., 2015; Vo et al., 2015). To propose an alternative way to tackle this problem, this chapter focuses on the analysis of neural activations to identify the degree of being visually perceptible, namely the *visualness* of a given attribute. The analysis takes advantage of the layered structure of the deep model in order to determine the semantic depth of the attribute.

3.1 Datasets

Since the visual attributes of objects have contextual dependency to the object domain, this dissertation studies the consistent meaning of the attributes in the clothing domain to avoid semantic shifts. Thus, two domain-specific datasets from an online e-commerce website and a social networking website were acquired.

3.1.1 Etsy dataset

The Etsy dataset is a collection of data acquired from the online market of handcrafted products, called *etsy.com*. Each product listed in Etsy contains an image, a title, a product description, and various metadata such as tags, category, or price. Considering the trade-off between dataset size and domain specificity, the product images under the clothing category that includes 247 subcategories (e.g., `clothing/women/dress`) were collected.

Near-duplicate removal

As is common with any Web data, the raw data acquired from Etsy contains a huge amount of near-duplicates. The major characteristics of the Etsy data include the following: 1) there are many shops, but the number of sold items per shop is exhibited by a long-tail. Meaning many shops sell only a few items, while only a few shops sell many items, and 2) many shops tend to sell similar items, e.g., the same black hoodie in the same background with a different logo patch, and in an extreme case, just a few words (proper nouns) being different in the product description. The near-duplicate removal is primarily designed to prevent such proper nouns from building up a category. The results show that without the near-duplicate removal, the system severely suffers from overfitting and resulting in meaningless results.

To remove near-duplicate data, the system applies the following procedure: 1) group product listings by shop, 2) computing a bag-of-words from a title and description for each item within the group except English stop words, 3) computing the cosine distance between all pairs of products, 4) applying agglomerative clustering by thresholding the pairwise cosine distance, 5) randomly picking one product from each cluster.

The duplicate-removal process was applied for all shops in the dataset while for each shop, any pairs of products that had less than 0.1 cosine distance are merged into the same cluster. After the near-duplicate removal process has been applied, the Etsy dataset contained 173,175 clothing products.

Syntactic analysis

Given the title and description of the products in the Etsy dataset, the system applies syntactic analysis (de Marneffe et al., 2006) and extracts part-of-speech (POS) tags for each word. In this dissertation, the 250 most frequent adjectives (`JJ`, `JJR`, and `JJS` tags) are considered potential attribute words. Unless noted, the following experiments use the (50%, 25%, 25%) splits of data for training, testing, and validation.

3.1.2 WEAR dataset

The WEAR dataset is a large collection of images acquired from the social fashion sharing website *wear.jp*, where each post contains an image, associated shots from different views, a list of items, blog text, tags, and other metadata. From the crawled data, a subset of 212,129 images was randomized for the experiments.

From the WEAR dataset, the user-annotated tags are treated as the candidate words. The majority of tags acquired from the WEAR dataset were not only in Japanese, but also included multiple synonyms treated as different tags and typos. To mitigate this problem, user-annotated tags were translated into English using Google Translate and manually fixed, and then merged if they correlate to the same English word. After translation, the most frequent 250 tags were picked as a set of attribute candidates for further experiments.

3.2 Attribute discovery

The attribute discovery framework begins by splitting the weakly-annotated dataset into positive and negative sets, then computes the Kullback-Leibler divergence (KL) for each activation unit in the deep neural network. The KL divergence is then used to determine the important neurons for the given attributes. The degree of being visually perceptible or the *visualness* of attributes can be estimated from these selected neurons.

3.2.1 Divergence of neural activations

Although the image representation (neural activations) from the deep network captures numerous discriminative features in an image (Zeiler and Fergus, 2014), each neuron only sparsely responds to visual stimuli. The first step of the proposed framework attempts to discover the neurons that respond highly to the visual pattern associated with a given attribute word by using the KL divergence of neuron activations. These highly responding neurons are further described as *prime units*.

The framework begins by splitting the dataset D into positive and negative sets according to the weak annotation (adjectives or tags in Sec 3.1). While positive sets D_u^+ contain images with the candidate attribute-word u, the negative set D_u^- is the opposite.

Using a pre-trained neural network, the empirical distribution of the neural activations from all of the units in the network was computed. Let P_i^+ and P_i^- denote the empirical distribution of the positive / negative set for each neuron i. P_i^+, P_i^- were computed from the max-pooling results over the spatial dimension in all channels of the convolutional layers. Finally, the symmetric KL divergence S_i of the word u for each activation unit i of the network is defined as:

$$
\begin{aligned}
S_i(u|D) &\equiv D_{\mathrm{KL}}(P_i^+||P_i^-) + D_{\mathrm{KL}}(P_i^-||P_i^+) \\
&= \sum_x P_i^+(x) \log \frac{P_i^+(x)}{P_i^-(x)} + \sum_x P_i^-(x) \log \frac{P_i^-(x)}{P_i^+(x)}, \qquad (3.1)
\end{aligned}
$$

where x is the activation of the unit corresponding to the histogram bins. The resulting KL divergence $S_i(u|D)$ serves as an indicator to find the prime units for the word u. The intuition is that if the word u is associated with specific visual stimuli, the activation pattern of the positive set should be different from the negative set and that should result in a larger KL divergence for highly visual attributes such as color (e.g., red, white) or texture (e.g., floral, stripped) than the less visual attributes such as expensive or handmade. In other words, the system should be able to identify the visual pattern associated with the given word by finding neurons with a higher KL divergence.

3.2.2 Visualness

According to the previous work (Berg et al., 2010), the *visualness* is defined as the classification accuracy given the balanced positive and negative sets:

$$V(u|f) \equiv \text{accuracy}(f, D_u^+, D_u^-), \tag{3.2}$$

where f is a binary classification function. To eliminate the bias influence, balanced samples are subsampled from both positive and negative sets D_u^+, D_u^-. The neural activations are used as a feature representation to build a classifier, and the KL divergence S_i is used as resampling and feature-selection criteria to identify important features for a given word u.

Selecting and resampling by activations

The noisy positive and negative sets D^+, D^- bring undesirable influences when evaluating the classification accuracy of the word Eq. (3.2). This dissertation proposes a system to learn a visual classifier in two steps: 1) learning the initial classifier based on the activations from the prime units and 2) learning the visual classifier from the confident samples of the initial classifier.

More specifically, the system begins by selecting the top 100 prime units according to the KL divergence Eq. (3.1); then, the activations from these units are used as a 100-dimensional feature to learn an initial classifier[1] using logistic regression (Fan et al., 2008) and to identify the confident samples for the second classifier.

[1] Gaussian Naive Bayes also works in this setting, but a stronger classifier such as SVM with RBF kernel tends to overfit.

Learning attribute classifier

Once an initial classifier is learned, the second step is to train the visual classifier from the confident samples ranked by initial classifier confidence. The logistic regression will be trained using the learned representation from all neural activations (9,568 dimensions). The final accuracy evaluation Eq. (3.2) on the balanced test set gives the visualness of the given word.

3.2.3 Human perception

To evaluate the proposed visualness of the attributes, the human perception or visualness is required. Inspired by the observation from (Parikh and Grauman, 2011), it is harder for humans to provide the absolute visualness score than the relative score. This involves similar intuition to the KL divergence process, which states that if the word u is associated with specific visual stimuli, it should be easy for humans to distinguish between the positive sample and the negative sample, which should result in a higher human agreement among the subjects for the highly visual attributes than the less visual attributes.

Thus, the Amazon Mechanical Turk (AMT) task is designed to collect the human judgment of visualness as follows: given a word, two images are shown to the annotators where one is from the positive set and the other is from the negative set. The annotators are then required to pick the image that is more relevant to the given attribute; otherwise, the answer is none. The 100 most frequent words from the Etsy dataset are selected for evaluation purposes. For each word, 50 pairs of positive and negative images are randomly sampled, and 5 annotators are required to complete an image pair annotation. Let $H(u)$ denotes the human visualness of word u as the ratio of positive annotator agreements:

$$H(u) \equiv \frac{1}{N} \sum_{k}^{N} 1 \left[h_k^+(u) > \theta \right] \tag{3.3}$$

where 1 is an indicator function, $h_k^+(u)$ is the number of positive votes for image pair k, N is the number of annotated images, and θ is considered the threshold. In this experiment, θ is set to 3 for 5 annotators. Finally, the correlation between the machine visualness ranking and the human perception ranking is used to evaluate the proposed approach.

3.2.4 Experimental results

Since the prime units in the Convolutional Neural Network (CNN) activates differently depending on the network, three different models are explored as follows:

- **Pre-trained:** Reference CaffeNet model (Krizhevsky et al., 2012) implemented in (Jia et al., 2014) and trained on the ImageNet 1000 categories.
- **Attribute-tuned:** A CNN is fine-tuned to directly predict the weakly-annotated words in the dataset, assuming they are the ground truth and the noise is ignored. The soft-max layer in the CaffeNet is replaced with a sigmoid to predict 250 words.
- **Category-tuned:** A CNN is fine-tuned to predict the 247 sub-categories of clothing using the metadata in the Etsy dataset, such as t-shirts, dresses, etc.

The basic AlexNet model has been chosen to evaluate how fine-tuning affects the attribute discovery task, however different architectures such as VGG (Simonyan and Zisserman, 2014) can be used to do the same task as well. The category-tuned model demonstrates the effect of domain transfer without being overfitted to the target labels. The following different visualness definitions are compared against human perception.

- **CNN+maximum KL-div:** To observe the correlation between visualness and the neural activations of the prime units (bypass two-step classifier), the visualness is defined as the largest KL divergence $S_i(u|D)$ across all layers.

- **CNN+random:** The random sub-sampling of the same number of positive and negative images so as to learn a logistic regression from all of the neural activations in the CNN, and to then use the testing accuracy to define the visualness. This is similar to the visualness prediction in the previous work (Berg et al., 2010), except that the neural activations are used as a feature.

- **CNN+initial**: Testing accuracy of the initial classifier trained on the most activating neurons or prime units.

- **CNN+resample:** Testing accuracy of the attribute classifier trained on the resampled images according to the confidence of the initial classifier and learned from all of the neural activations, as has been described in Sec 3.2.2

- **Attribute-tuned:** Average precision of the direct prediction of the Attribute-tuned CNN in the balanced test set.

Table 3.1: Visualness correlation to human perception.

Method	**Feature dim.**	**Pearson**	**Spearman**
Pre-trained+maximum KL-div	-	0.672	0.527
Pre-trained+random (baseline)	9,568	0.737	0.637
Pre-trained+initial	100	0.760	0.663
Pre-trained+resample	9,568	**0.799**	0.717
Attribute-tuned+maximum KL-div	-	0.575	0.455
Attribute-tuned	4,096	0.662	0.549
Attribute-tuned+random	9,568	0.760	0.684
Attribute-tuned+initial	100	0.663	0.480
Attribute-tuned+resample	9,568	0.783	0.704
Category-tuned+maximum KL-div	-	0.665	0.489
Category-tuned+random	9,568	0.716	0.565
Category-tuned+initial	100	0.716	0.603
Category-tuned+resample	9,568	0.782	**0.721**
Language prior	-	0.139	0.032

- **Language prior:** The n-gram frequency of adjective-noun modification for the given attribute-word from the Google Books N-grams (Michel et al., 2010). The language prior is considered as a reference to understand the scenario when visual data is not accessible. The assumption is that for each of the object categories in Etsy, the visual modifier should co-occur more than the non-visual words. For example, for *dress* category, the words *'floral dress'* or *'white dress'* should appear more than *'available dress'* or *'expensive dress'* in the literature (Michel et al., 2010). The prior is computed using the sum of n-gram probability on attribute-category modification to 20 nouns in the Etsy clothing categories.

Quantitative evaluation

Table 3.1 summarizes the Pearson and Spearman correlation coefficients to human perception using different definitions of visualness together with the feature dimensions for each approach. From the table, the maximum KL-div results confirm that there exists a correlation between the KL-divergence of neural activations from the prime units and the visualness of the attributes, showing as a positive correlation with human perception.

Table 3.2: Most and least visual attributes discovered in Etsy dataset.

Method	**Most visual**	**Least visual**
Human	flip pink red floral blue sleeve purple little black yellow	url due last right additional sure free old possible cold
Pre-trained+resample	flip pink red yellow green purple floral blue sexy elegant	big great due much own favorite new free different good
Attribute-tuned	flip sexy green floral yellow pink red purple lace loose	right same own light happy best small different favorite free
Language prior	top sleeve front matching waist bottom lace dry own right	organic lightweight classic gentle adjustable floral adorable url elastic super

Moreover, the results show that even though the initial classifiers are only learned through the filtering of 100-dimensional feature using the prime units, the higher Spearman correlation to human perception is achieved than the random baselines with a 90-time larger feature. Resampling images by the initial classifier confidence improves the correlation to human perception over the random baseline in all models. These results demonstrate that feature-selection and resampling using the high-KL neurons helps the discovery of visual attributes in the noisy dataset.

In addition, the result suggests that directly fine-tuning against the noisy annotation (attribute-tuned) can harm the representational ability of neurons since fine-tuning to domain-specific data with possibly non-visual words can lead to overfitting and the suppression of neuron activity in the network even if they are important for recognition. Indeed, the alternative fine-tuned model (category-tuned) gives a slightly better human correlation. The pre-trained network gives a slightly higher Pearson correlation. One explanation is because the neurons are trained on a wider range of visual stimuli in the ImageNet than in a domain-specific dataset like Etsy, which helps reproduce human perception. The low correlation from language prior indicates the difficulty of detecting visual attributes from only textual knowledge.

Qualitative evaluation

Table 3.2 lists the most and the least visual attributes for some of the selected methods. Note that the error in syntactic analysis incorrectly marked some nouns as adjectives, such as *url* or *flip* (*flip-flops*) here. Generally, CNN-based methods result in a similar choice of the most visual words such as colors (e.g., *pink*, *red*, *purple*, etc.) or texture (e.g., *floral*, *lace*, etc.). Unlike the most visual words, many least visual words have a similar visualness (almost zero or zero), thus the 10 least visual words across all methods that are shown in Table 3.2 (the right most column) are diverse. The language prior involves picking very different vocabulary due to the lack of visual clue in Google Books. For example, *'matching (couple) t-shirt'* is very common in the textual domain; however, the word *'matching'* is non-visual.

Figure 3.1 shows examples of the most and the least confident images according to the pre-trained+resample model. From concrete concepts like *orange* to more abstract concepts such as *elegant*, the results confirm that the automatic approach can learn various attributes from the noisy dataset. Figure 3.2 shows examples of the most and least *floral* images from both the positive and negative sets. As seen in the figure, the noise in the dataset introduces a lot of true-negatives (not mentioned but actually a floral product) and false-positives (floral is mentioned in the text but not relevant to the product). The automatically learned attribute classifiers can function as a dataset purifier for a noisy dataset.

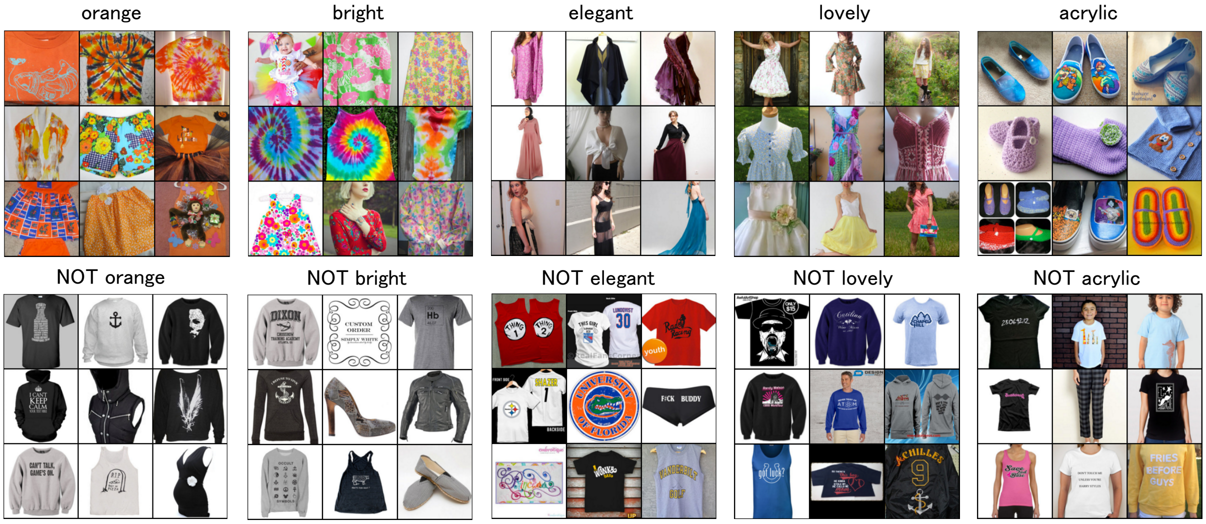


Figure 3.1: Examples of most and least predicted images for some of the attributes.

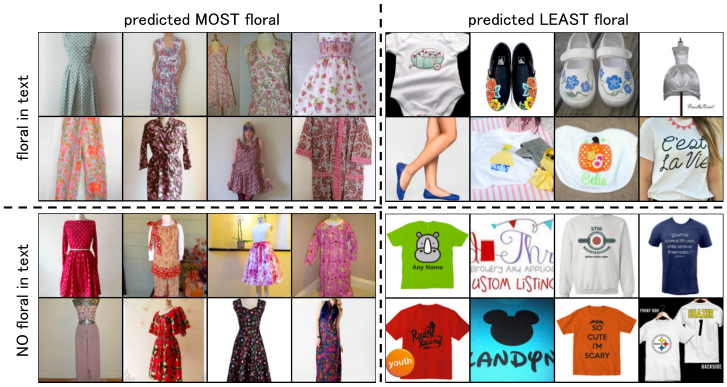


Figure 3.2: Most and least *floral* images. With the automatically learned classifier, the true-negatives and false-positives in the dataset can be discovered.

3.3 Understanding perceptual depth

This section explores how each layer in the neural network relates to attributes. It is well-known that neurons in a different layer activate different types of visual patterns (Zeiler and Fergus, 2014; Escorcia et al., 2015). This section is a further attempt to understand what type of semantic concepts directly relate to neurons using the KL divergence.

Consider the activation with respect to the layer depth, then the relative magnitude of max-pooled KL divergence for the layer l as:

$$S_l(u|D) \equiv \frac{1}{Z} \max_{i \in l} S_i(u|D), \text{ where } Z \equiv \sum_l \max_{i \in l} S_i(u|D). \tag{3.4}$$

The system is able to identify the most salient words by ranking the attribute vocabulary based on $S_l(u|D)$.

Tables 3.3 - 3.4 list the most salient 10 words for each layer of CNN in the Etsy and the WEAR datasets from the pre-trained CaffeNet. From both tables, the more primitive visual concepts like color words (e.g., *orange*, *green*) appear in the earlier stages of the CNNs, and as one moves down the network towards the output, the more complex visual concepts are observed. The same trend appears from both the Etsy and the WEAR datasets, even though the two datasets are very different (Etsy images are clothing images on a clear background while WEAR images are street fashion: outfits on people) and both are considered noisy (missing or incorrect descriptions/tags).

Note that there are non-visual words in a general sense due to the dataset bias. For example, *genuine* in Etsy tends to appear in the context of the phrase *genuine leather*, and the word *many* appears in the context of *many designs* available for sweatshirt products. An example of such a dataset bias results in higher divergence of the neuron activity. One approach to deal with such context-dependency is to consider the phrase instead of a single adjective since the words *genuine* and *many* have been identified as non-visual terms when they stand alone, while *genuine leather* and *many designs* are considered

norm1	**norm2**	**conv3**	**conv4**	**pool5**	**fc6**	**fc7**
orange	green	bright	flattering	lovely	many	sleeve
colorful	red	pink	lovely	elegant	soft	sole
vibrant	yellow	red	vintage	natural	new	acrylic
bright	purple	purple	romantic	beautiful	upper	cold
blue	colorful	green	deep	delicate	sole	flip
welcome	blue	lace	waist	recycled	genuine	newborn
exact	vibrant	yellow	front	chic	friendly	large
yellow	ruffle	sweet	gentle	formal	sexy	floral
red	orange	french	formal	decorative	stretchy	waist
specific	only	black	delicate	romantic	great	american

Table 3.3: Most salient words from the Etsy dataset for each CaffeNet layer.

visual.

How fine-tuning affects perceptual depth

Fine-tuning has an influence on the magnitude of the layer-wise max-pooled KL divergence in that 1) the pre-trained model activates almost equally across layers and 2) the category-tuned model induced larger divergence in the mid-layer (conv4), while 3) the attribute-tuned model activates more in the last layer (fc7). Figure 3.3 shows the relative magnitude of the average layerwise max-pooled KL divergence:

$$M_l \equiv \frac{1}{|U|} \sum_{u \in U} \sum_{i \in l} S_i(u|D) \tag{3.5}$$

The attribute-tuning causes a direct change in the last layer as expected, whereas the category-tuning brings a representational change in the mid-layers. The results suggest that the domain-specific knowledge is encoded inside the activations from the mid-to-higher layers, but there are domain-agnostic features in the earliest layers which are useful for recognizing primitive attributes such as color. Moreover, the set of salient

norm1	**norm2**	**conv3**	**conv4**	**pool5**	**fc6**	**fc7**
blue green red-black red denim-on-denim denim-shirt pink denim yellow leopard	denim-jacket pink red red-socks red-black champion blue white shirt i-am-clumsy yellow	border-striped-tops border-stripes dark-style stripes backpack red dark-n-dark denim-shirt navy outdoor-style	kids bucket-hat hat-n-glasses black sleeveless american-casual long-cardigan white-n-white stole mom-style	shorts half-length pants denim dotted border-stripes white-pants border-tops gingham-check sandals chester-coat	white-skirt flared-skirt spring upper beret shirt-dress overalls hair-band loincloth-style matched-pair	long-skirt suit-style midi-skirt gaucho-pants handmade straw-hat white-n-white white-coordinate white-pants white

Table 3.4: Most salient words from the WEAR dataset for each CaffeNet layer.

words per layer stays similar after fine-tuning in either case: earlier layers activate more on primitive attributes, color or texture, and later layers activate more on abstract words.

How each layer relates to human perception

This section evaluates how each layer relates to human perception, using the annotation from Sec 3.2.3. Figure 3.4 plots the Pearson correlation of the layer-wise maximum KL divergence Eq. (3.4) against human visualness. The plot suggests that the activation of mid-layers is closer to the human visualness perception, but interestingly, the last fully-connected layers give a negative correlation. It is positive that the last layers are more associated to abstract words that are not generally considered visual by humans. However, they are contextually associated to visual patterns in the domain-specific data.

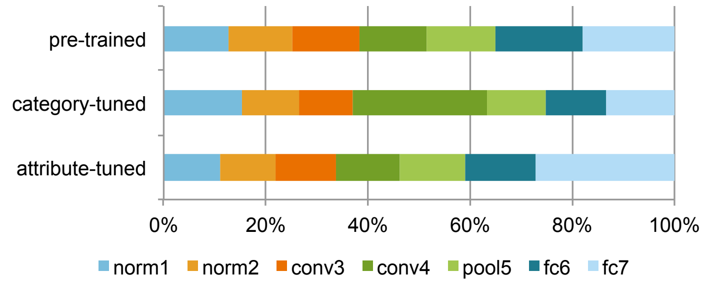


Figure 3.3: Relative magnitude of average layer-wise maximum KL divergence.

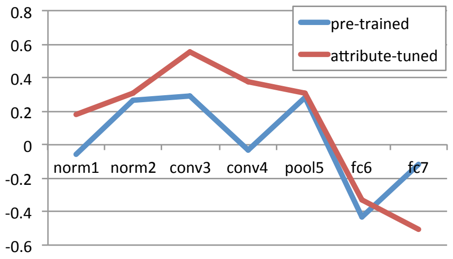


Figure 3.4: Pearson correlation coefficients between human visualness and max KL divergence of each CNN layer.

3.4 Saliency detection

Convolutional neural networks demonstrate their benefit in many tasks including class-specific saliency detection (Simonyan et al., 2014; Zhou et al., 2014). To emphasize the advantage of the neural activations from prime units, this dissertation proposes an approach which detects the salient regions in the given image directly from the neurons are highly-activated.

3.4.1 Cumulative receptive fields

This section introduces the saliency detection with respect to the given attribute based on the receptive field (Zhou et al., 2014). The main idea is to accumulate the neurons' response in order of the largest KL divergence to the least. The detection pipeline starts as follows:

1. Applying the sliding-windows of a multi-scale occluder to the input image. In the experiments, the occluder sizes include 24×24, 48×48, and 96×96 with a stride size 4 for the 256×256 input image.

2. Forwarding the occluded images through the CNN.

3. Observing the differences in activations as a function of the occluder location $a_i^j(x, y)$ for unit i at occluder size j.

4. Applying a Gaussian filter with the scale proportional to the occluder size j of the response map $a_i^j(x, y)$.

5. Applying an average pooling over the response maps from multi-scale occulder j to generate a single response map $A_i(x, y)$.

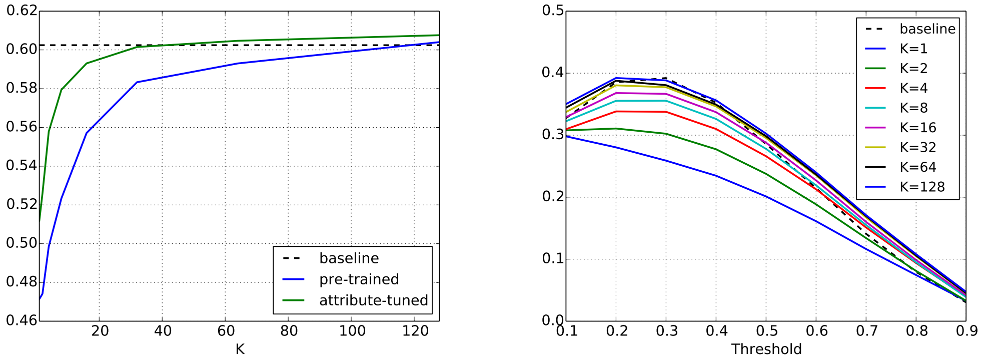
The resulting response map $A_i(x, y)$ can have either positive or negative peaks to the input pattern. The system heuristically negates and inverts the response map if the map has negative peaks. The response map of unit i is then normalized to be within $[0, 1]$ scale by $R_i(x, y)$. The final saliency map M given image I and word u is computed by accumulating the units ordered and weighted by the KL divergence:

$$M(x, y|u, I) \equiv \frac{1}{Z} \sum_i^K S_i(u|D) R_i(x, y|I), \tag{3.6}$$

where $Z = \sum_i^K S_i(u|D)$. The units are accumulated by the largest unit divergence $S_i(u|D)$ up to K.

3.4.2 Human annotation

Since the images in the Etsy dataset consist mostly of a single object appearing in the center of the image frame and localization is merely needed, the WEAR dataset is used



(a) Mean AP

(b) Mean IoU

Figure 3.5: Saliency detection performance in terms of (a) mean average precision and (b) mean IoU of the attribute-tuned model over the heat-map threshold. Accumulating receptive fields by KL improves the detection performance, and even the pre-trained model can reach the baseline performance without any fine-tuning.

for saliency evaluation. Similarly to Sec 3.2.3, human annotation on the salient regions is required for evaluation purposes. For the randomly selected set of 10 positive images of the most frequent 50 tags in the WEAR dataset, 3 annotators are required to draw bounding boxes around the relevant region to the specified tag-word. The pixels which have 2 or more annotator votes are counted as the ground-truth salient regions. The images that have no worker agreement are discarded from the evaluation.

3.4.3 Experimental results

Figure 3.5(a) plots the average detection performance from all the tags in terms of the mean average precision (mAP) for predicting pixel-wise binary labels, and the mean intersection-over-union (IoU) of the attribute-tuned model as shown in Figure 3.5(b). The results show the IoU for the binarized saliency map $M(x, y|u, I) \geq \theta$ at the different threshold θ.

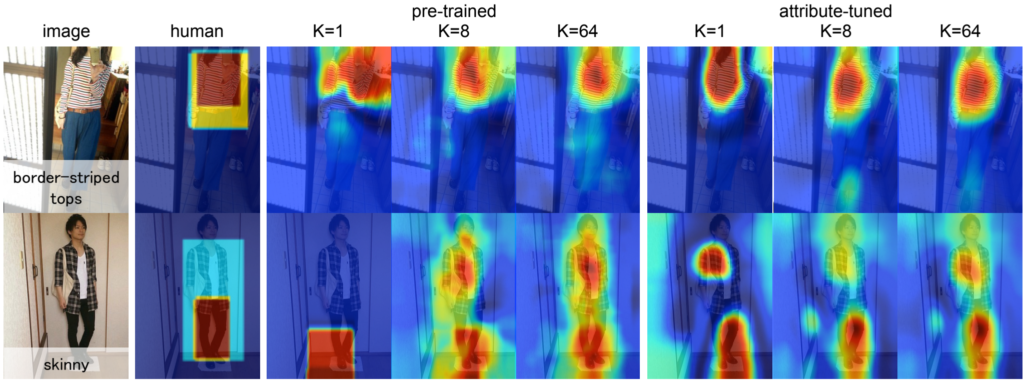


Figure 3.6: Accumulating receptive fields by the largest KL divergence. As adding more neurons, the saliency heat-map becomes finer.

Both plots from Figure 3.5 show the performance with respect to the number of accumulations K, as well as the baseline performance of the smoothed gradient magnitude (Simonyan et al., 2014) of the attribute-tuned model. The results show that the detection performance improves as more neurons are accumulated in the saliency map according to the divergence, and gives on par or slightly better performance against the baseline. Note that even the pre-trained model can already reach the baseline by this simple accumulation based on KL divergence, without any optimization towards saliency. The improvement in both the pre-trained and attribute-tuned models is observed, but the pre-trained model tends to require more neurons. One explanation is that fine-tuning makes each neuron activate more to a specific pattern while reducing activations on irrelevant patterns, which then results in the diminishing accumulation effect. The results also suggest that visual attributes are combinatorial visual stimuli rather than some visual pattern detectable only with a single neuron since the larger K leads to the better mean AP and mean IoU.

Figure 3.6 shows the detection results by human annotation and by the cumulative

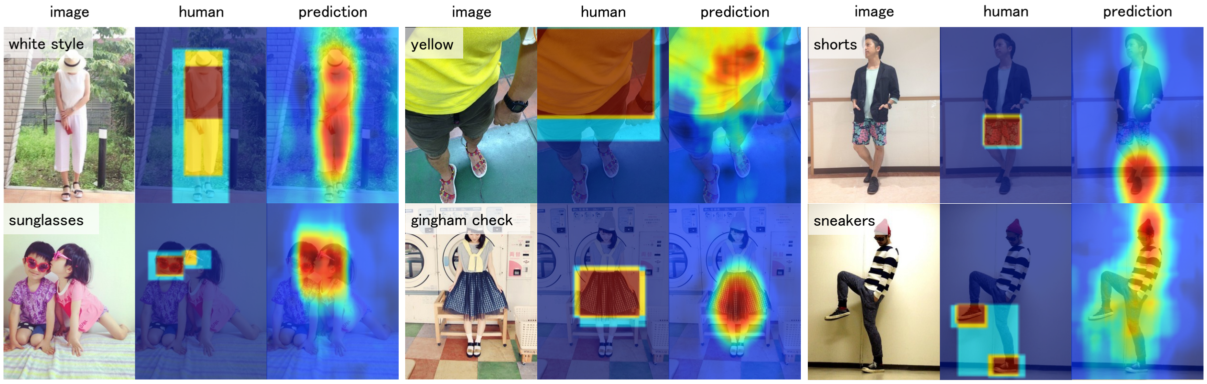


Figure 3.7: Results of detected salient regions for the given attribute. The rightmost column shows failure cases due to distracting contexts or visibility issues.

receptive field using a CNN pre-trained on ImageNet or fine-tuned on the WEAR tags, when the accumulation size K is 1, 8, and 64.

Figure 3.7 shows the results of human annotation and the pre-trained CNN with the accumulation size $K = 64$ where saliency detection methods using a pre-trained CNN perform remarkably well *even without fine-tuning*. As more neurons are accumulated, the response map tends to produce a finer degree of localization.

Accumulation helps most of the cases, but failure cases are spotted when there is a distractor co-occuring with the given attribute. For example in Figure 3.7, the detection of shorts fails because legs always appear with shorts and the system ends up with the leg detector instead of a shorts detector (distractor issue). Moreover, the proposed method tends to fail when the target attribute is associated to only a small region in the image (visibility issue).

3.5 Conclusion

This chapter has shown that it is possible to discover and analyze new visual attributes from noisy Web data using neural activations. The key idea is the use of neurons that are highly activated in the network, and that are identified by the KL divergence of their activation distribution in a weakly annotated dataset. The empirical study using two real-world datasets demonstrates that the proposed approach can automatically learn a visual attribute classifier that has a perceptual ability that is similar to humans. Consequently, the depth in the network relates to the depth of attribute perception and the neurons can detect salient regions in the given image.

CHAPTER 4: RECOGNITION AND ANALYSIS OF VISUAL STYLE TRENDS

This chapter explores more informative attributes, focusing on visual styles. The visual styles are more challenging because they correspond to multiple visual representations. In some domains like fashion, *1900s* corresponds to *corset, broad ribbon tie, lace collar, pouter-pigeon shape, frilly blouses,* etc. The visual styles are crucial for many tasks. By identifying the visual styles of each era, it is feasible to explore several interesting problems such as dating historical photographs and temporal classification for data organization. Especially, in the fashion domain where the reoccurrence of visual concepts has been observed from time to time or visual trends, the visual styles can lead to certain appealing applications such as trend analysis and prediction problem.

Therefore, this chapter starts by presenting the deep learning methods employed for estimating when objects were made based on their visual appearances. Toward this goal, the first proposed method utilizes features from existing deep networks. Then, the deep networks are fine-tuned for the purpose of dating an object. The results show that the deep learning method outperforms a color-based baseline and significantly improves on the previous state of the art for dating historical objects. Unlike hand-crafted representations, the learned representations from deep networks are not immediately interpretable. Thus, the analyses of the neural activations and their entropy are provided in order to gain an additional level of understanding about the deep network. While the direct applications of dating historical objects framework include large-scale data organization or image retrieval, the framework can be applied to analyze the influence of vintage fashion on the runway collections and finally to analyze the influence of runway collections on street fashion.

4.1 Datasets

This chapter includes five different datasets: 1) The Car Database (Lee et al., 2013), 2) A large novel collection of clothing photographs with associated dates, collected from *Flickr.com*, 3) Manually annotated clothing photographs from museum collections, 4) A set of large fashion show collections acquired from *Vouge.com*, and finally 5) The Paper Doll dataset (Yamaguchi et al., 2013) which is a collections of street fashion collected from *chictopia.com*.

Car Database (CarDb): The Car Database (Lee et al., 2013) contains 13,474 photographs of cars made from 1920 to 1999 resulting in 8 temporal classes collected from the *cardatabase.net*.

Flickr Clothing Dataset: The system first acquired hundreds of thousands of images from a wide variety of 50 Flickr groups focus on vintage fashions, e.g., *Fashions Past - Best and Worst* and *As She Was*. A face detection algorithm (Zhou et al., 2013) is then applied to automatically filter out images without a depicted person. After that, the images are manually inspected to remove additional non-photographic content such as artwork, painting, and advertisements. To automatically assign the temporal label to an image, the meta-data such as title, description, and tags are taken into account. From the text information, the temporal word that appears the most will be assigned as temporal label for an image. Finally, the dataset will contain 58,350 clothing photographs with corresponding meta-data, including photo id, user id, title, description, tags, longitude, latitude, number of views and groups, etc.

Museum Dataset: Since the user annotated date in Flickr can sometimes be noisy, additional photographs have been collected from museum collections. The museum dataset contains vintage photographs labeled by expert museum curators from 2 different museums; the Metropolitan Museum of Art, and Europeana Fashion. The Museum dataset contains 9,421 images taken between 1900 and 2009, showing the clothing that was worn on people. The dataset is treated as an alternative test set to evaluate clothing date models trained on the larger Flickr clothing dataset. As this Museum dataset has a different domain than the Flickr images, it also can be used to evaluate model generalizations, i.e. by training on one dataset and testing on another.

Runway Dataset: This dataset consists of 348,598 images taken from 9,328 fashion shows involving a wide variety of brands over a period of 15 years, from 2000 to 2014, that were taken from *Vouge.com*. The metadata includes season (e.g., Spring 2014), category (e.g., Ready-to-Wear, Couture), brand name, date of the event, and a short text description. In the dataset, there are 852 distinctive brand names, ranging from haute couture designers like Chanel or Fendi, to more common consumer brands like J Crew or Topshop. Most brands have between 10 to 100 photos, while a few brands have significantly more. Note that in this dataset, season refers to a specific fashion event (e.g., fashion week), which might be different from the date of the event. The most common events is the Ready-to-Wear shows that are held in the spring and fall.

Paper Doll dataset: The Paper Doll dataset (Yamaguchi et al., 2013) is used to sample street photos of outfits that regular people wear everyday. This dataset contains 339,797 images collected from *chictopia.com*. The street photographs from the Paper Doll dataset is used as the test set for cross-domain retrieval system to study how runway fashions influence everyday fashions on the street.

4.2 Dating historical objects approaches

This chapter begins by proposing the deep learning approaches that are used for estimating when objects were made. Since the deep network shows the remarkable performance in several tasks, both the pre-trained network and fine-tuning approach are explored in this section.

4.2.1 Pre-trained + Classifier

Two Convolutional Neural Network (CNN) models pre-trained on 1.2million labeled images from ImageNet; AlexNet (Krizhevsky et al., 2012) and VGG (Simonyan and Zisserman, 2015) have been used as the feature extraction unit in this experiment. For each network, the learned representation from the second fully-connected layer is computed and used as a visual representation to train two different classification model: a linear Support Vector Machines (Fan et al., 2008) with fixed $C_{svm} = 0.1$ as used in the color based approach (Palermo et al., 2012), and Support Vector Regressors (Chang and Lin, 2011) with fixed $\epsilon = 0.1$ and set $C_{svr} = 100$ as used in the data-driven approach (Lee et al., 2013).

4.2.2 Fine-tuning (FT)

From each pre-trained model (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015), three different fine-tuning models have been explored as follow:

- **CarDb:** Fine-tuning a CNN on 10,130 training images and tested on 3,343 images from the CarDb using the same train/test split as the data-driven baseline (Lee et al., 2013).
- **Clothing:** Fine-tuning a CNN using 3/4 of images from the Flickr clothing dataset.
- **Black/White (BW):** Since the Flickr clothing and Museum datasets depict vintage photographs taken from 1900-2009, there exist the historic color cues (e.g., black and white color or sepia tones in early photos and color photographs in later photos) in both datasets. In order to constrain the network to learn about the temporal information of objects, without taking ”trivial” shortcuts, a CNN is fine-tuned using the same train/test split as the Clothing model but in black and white. To emphasize this experiment, the same experiment has been applied for the CarDb model as well. Note that the test images are also in black and white color.

To fine-tune the network, the last fully connected layer has been modified from 1000 classes to 11 temporal classes (one per decade) of the Flickr clothing dataset and 8 temporal classes for the CarDb. The models are trained for 50,000 cycles using stochastic

	CarDb	**Clothing**	**Museum**
Data-driven baseline (Lee et al., 2013)	8.56	17.74	19.56
Color based baseline (Palermo et al., 2012)	-	17.21	21.21
AlexNet + SVM	7.77	12.99	17.33
AlexNet + SVR	8.10	16.73	22.00
VGG-16 + SVM	6.90	12.60	16.35
VGG-16 + SVR	7.43	15.87	18.76
AlexNet (FT) + BW	6.78	17.16	17.96
AlexNet (FT)	6.17	12.88	16.43
VGG-16 (FT) + BW	4.27	13.66	16.40
VGG-16 (FT)	**3.97**	**11.54**	**14.23**

Table 4.1: The mean absolute error (years) training and testing on CarDb and training on the Flickr clothing dataset and testing on the held out clothing and Museum dataset.

gradient descent with a batch size of 50 examples, a momentum of 0.9, a weight decay of 0.0005 and a decrease the learning rate of the models to 0.00001.

4.3 Experimental results

Table 4.1 shows the mean absolute error (MAE) in years across the features and classifiers, including comparisons to the baselines (Palermo et al., 2012; Lee et al., 2013) on all datasets.

4.3.1 Pre-trained + Classifier

Without any modification of the pre-trained network, the deep feature already outperforms the previous state-of-the-art tests on both datasets. Moreover, for clothing, the results show that even though the domain shift is quite evident (the results of the Museum dataset are worse than the results of the held out portion of the Flickr clothing dataset), the deep learning feature is beneficial for the dating of vintage objects. The evaluations achieve error reductions of 0.95 ± 2.47 years and 3.19 ± 2.06 years on the

Museum and Flickr clothing collections respectively when compare to the baseline (Lee et al., 2013).

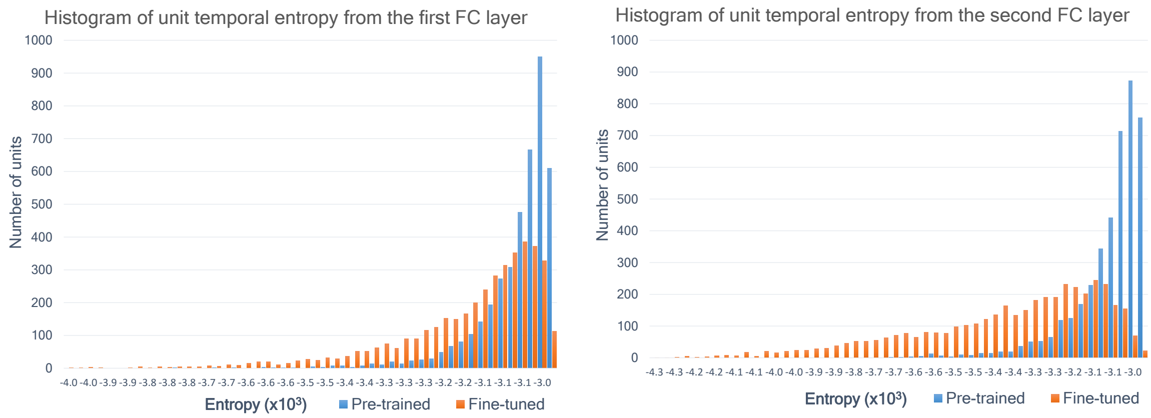
4.3.2 Fine-tuning (FT)

The fine-tuned models consistently outperform the pre-trained models on both object categories. For cars, the MAE decreases by around 2.48±0.86 years. Similar trends apply for clothing, wherein the fine-tuned model decreases MAE by around 2.86 ± 2.42 years on the Museum dataset and 1.67 ± 1.94 years on the nosier Flickr clothing dataset.

Moreover, the results confirm that the fine-tuned network learns visual elements beyond color by outperforming two color-based baselines. The first baseline (Palermo et al., 2012) proposed temporally discriminative features related to the evolution of color imaging processes over time, achieves the worse MAE on both datasets compare to the fine-tuned networks. The results from the fine-tuning network using using black/white images from the Flickr clothing dataset show that the B/W model achieves about 2.53±1.2 years higher MAE than the color model. These results emphasize that even though color is an important clue, the fine-tuned network is able to learn the temporally sensitive features of an object beyond color.

4.4 Deep Network Analyses

Based on the quantitative results, the deep learning methods appear to be promising for dating vintage objects. However, unlike the patch discovery methods proposed in some previous work (Lee et al., 2013), the learned representations from deep networks



(a) first FC layer

(b) second FC layer

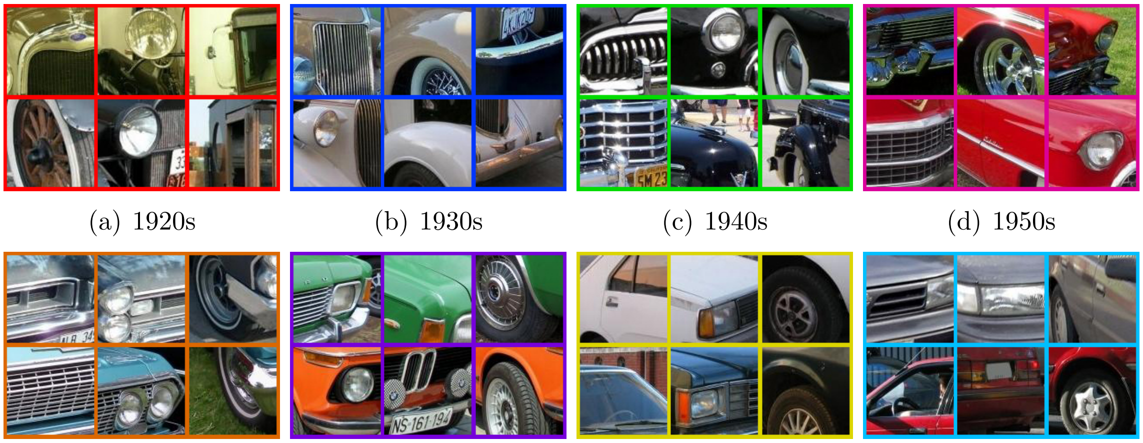
Figure 4.1: Histogram of unit temporal entropy from 2 different fully connected layers of pre-trained(blue) and fine-tuned network(orange).

are not immediately interpretable. Thus, this section provides some analyses of the CNN networks on the CarDb to gain additional understanding about what the fine-tuned networks have learned.

4.4.1 Temporally-sensitive units

Since the fine-tuned network outperforms the pre-trained network, it is reasonable to hypothesize that the temporal sensitivity of the neurons in the network has changed during the fine-tuning process. To verify this hypothesis, for each unit, the following procedures were applied: 1) rank images by their maximum activation, 2) quantize top $N = 500$ maximum activation images into a temporal histogram, by decade, 3) compute the entropy of each histogram as:

$$E(u) = -\sum_{i=1}^{n} H(i) \cdot log_2 H(i) \tag{4.1}$$



(a) 1920s (b) 1930s (c) 1940s (d) 1950s

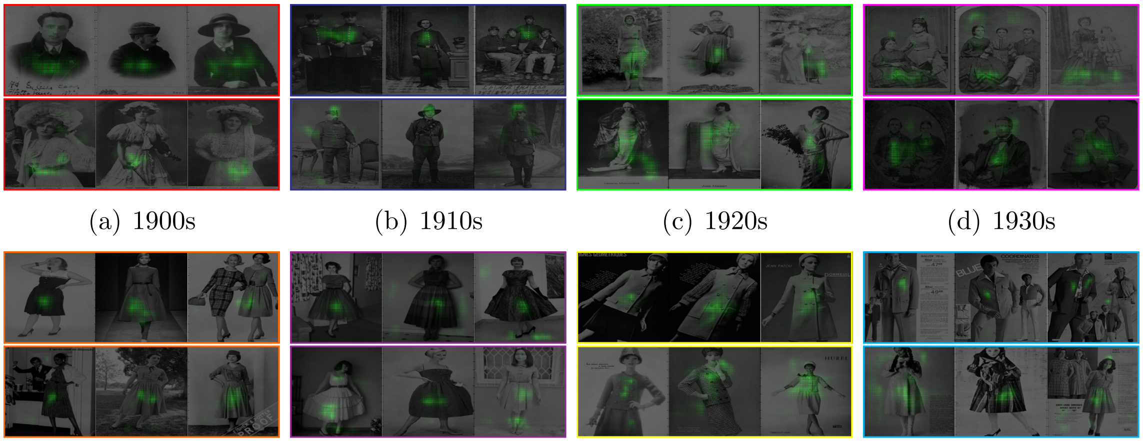
(e) 1960s (f) 1970s (g) 1980s (h) 1990s

Figure 4.2: In each rectangle, 3 image regions, from the 1920s to the 1990s, with the maximum activations from 3 different units from the 2nd FC layer of the fine-tuned network on the CarDb are shown.

where $H(i)$ denotes the histogram count for bin i and n denotes the number of quantized label bins. Note that lower entropy values indicate higher temporal sensitivity, and 4) compute the entropy histogram of all units from the fully-connected layers as shown in Figure 4.1(a) (first FC layer) and Figure 4.1(b) (second FC layer). Both histograms show that low entropy units appear more in the fine-tuned network than the pre-trained network. The results indicate that the units have been tuned to capture a temporally discriminative feature for a specific time period. This trend is visible in both layers, but is more pronounced in the second layer, which makes intuitive sense since this layer is closest to the classification layer.

4.4.2 Unit activation analysis

The results from Sec 4.4.1 demonstrate that the units have been tuned to capture the temporal sensitive elements for a specific time period. Thus, this section aims to reveal



(a) 1900s

(b) 1910s

(c) 1920s

(d) 1930s

(e) 1940s

(f) 1950s

(g) 1960s

(h) 1970s

Figure 4.3: Each rectangle shows 3 images with the maximum activation regions (in green highlighting) of the units which have the lowest entropy in each decade of the fine-tuned network on the Flickr Clothing dataset.

those visual elements and answer the question whether or not the temporal sensitive elements correspond to the semantic elements of objects.

To visualize the visual elements that have been captured by the network, the system follows the approach proposed by the previous work (Zhou et al., 2014) to estimate the receptive fields (RFs) of the units. To estimate a unit's RF, images are ranked by their maximum activations for that unit, and the top K images were selected to identify image regions that are highly activated. To recover the high activation regions within an image, each image is replicated many times with a small occluder of size 11x11 placed at one of about 5,500 locations in a dense grid (stride 3 pixels) in the images. Each occluded image is evaluated using the same network and the change in activation versus the original value is calculated. Those differences are combined into a discrepancy map over the image. The intuition behind this approach is that if there is a large discrepancy between activation values before and after occlusion, then the occluded region is important for activating that unit.

In order to investigate the most informative regions in an image, the system focuses on the image regions that highly contribute to the prediction decision. Only the true positive images from the fine-tuned network are included in this experiment. For a given image, the top N units were selected based on their contribution to the prediction decision. Finally, a discrepancy map of assigned images is computed following the same work (Zhou et al., 2014). Figures 4.2 - 4.3 show image regions that caused the maximum activation for the given unit from the last FC layer from the CarDb and Flickr clothing datasets. The results indicate that the networks have been tuned to the temporal sensitive elements which highly correspond to certain parts of the object such as front bumpers, headlights, or wheels for cars and cinched-in waists (40s-50s), mod dresses (60s) and leisure suits (70s) for clothing.

4.4.3 Discriminative part correlation

So far, the results reveal that during fine-tuning, the units in the network have been tuned to temporal sensitive elements and these visual elements highly correlate to object parts. These results lead us to an interesting question: Do these visual elements correspond to the style-sensitive elements discovered by the data-driven approach (Lee et al., 2013)?

To identify the correspondence between the visual elements learned by the fine-tuned network and the style-sensitive elements proposed by the data-driven approach (Lee et al., 2013), the pipeline starts by searching for units which have a similar behavior as

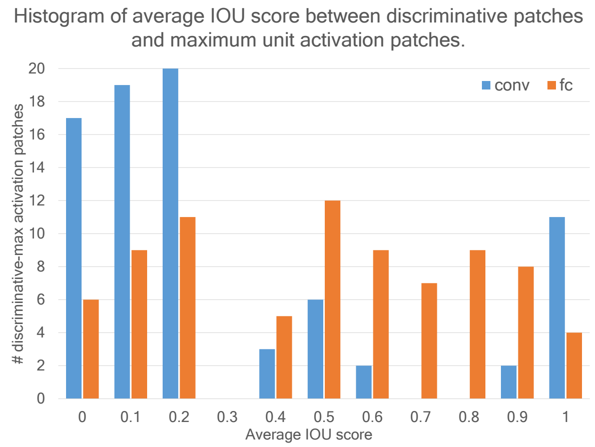


Figure 4.4: The average IoU score between style-sensitive patches and maximum activation patches.

their style-sensitive detectors where: (1) high responses on similar sets of images, and (2) similar localization patterns. To target these units two image rankings have been computed; the first one is based on the maximum activation for a given unit u over the images, and the other is based on the maximum detector confidence for a given detector d over images. Let $C(u, d)$ denote the correlation C between unit u and the generic detector d as $C(u, d) = \frac{|A_n \cap D_n|}{n}$ where A_n represents the set of top n images from the activation based ranking and D_n represents the set of top n images from the detection based ranking ($n = 30\%$ in all experiments).

The average correlation scores between the style-sensitive detectors and their top 5 correlated units from the last convolutional layer (conv) and a second fully connected layer (fc) are 0.521 and 0.543, respectively, which indicates that about half of the units in both layers overlap with style-sensitive detectors. In addition, the average Intersection-over-Union score (IoU) between the maximum activation patches and the style-sensitive patches are shown in Figure 4.4. These results emphasize that units in the network

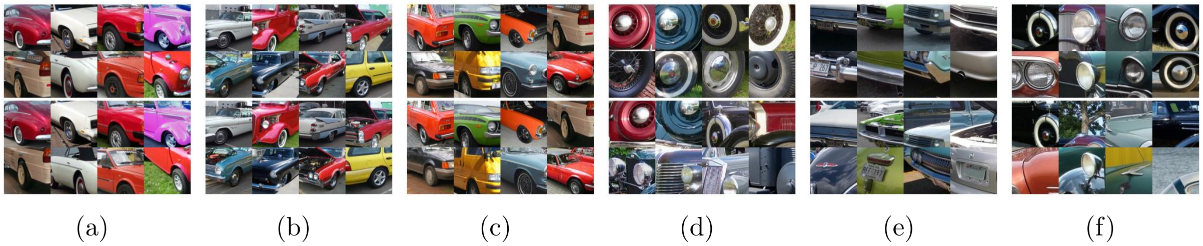
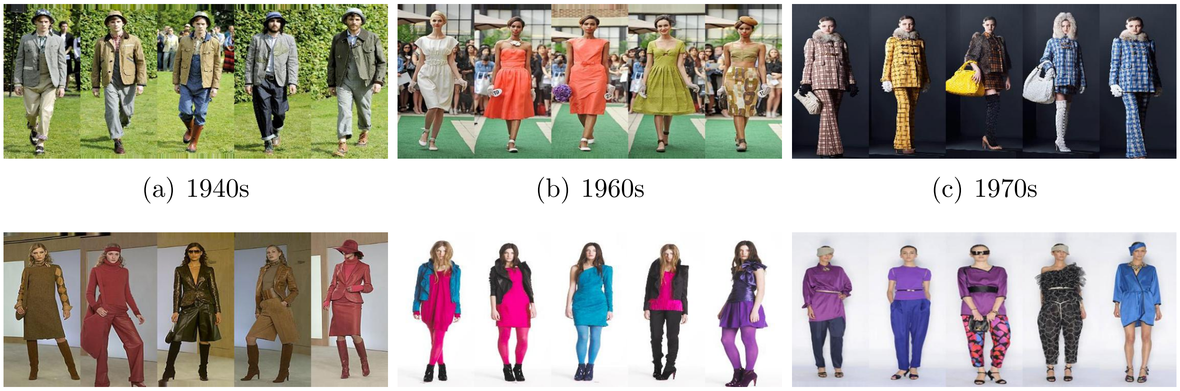


Figure 4.5: For each block, the top two rows show style-sensitive patches from the baseline (Lee et al., 2013), while the bottom two rows show regions with the maximum activation from the same images. While (a-c) shows unit activations which are highly correlated to style-sensitive patches, (d-f) shows unit activations which are poorly correlated to style-sensitive patches

are fine-tuned to temporally sensitive parts of an object. Additionally, only 7.5% of the patches revealed an average IoU score of less than 0.1 while 61.25% of the patches revealed an average IoU score of more than 0.5, confirming that the style-sensitive parts acquired from the baseline (Lee et al., 2013) have been automatically discovered by the network. Qualitative examples of high/poor correlation patches are shown in Figure 4.5. Although Figures 4.5(d) - 4.5(f) show visual elements which have low correlation to the baseline (Lee et al., 2013), these additional patches could be posited as being temporally sensitive, and contribute to the improved performance.

4.5 Analyzing the influence of vintage fashion

When thinking about fashion trends: denim miniskirts, ripped distressed jeans, tracksuits, preppy polo shirts with popped collars, neon colors, gladiator shoes, etc. Children look at their parents' vintage photographs and say "That's ridiculous! What were they wearing?". However, these trends keep coming back with some tweaks. Designers seek for the inspiration from everywhere, and that usually include the past!



(a) 1940s

(b) 1960s

(c) 1970s

(d) 1970s

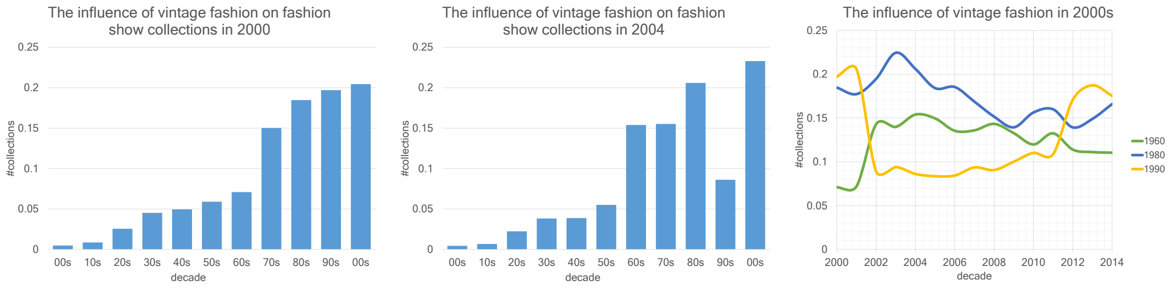
(e) 1980s

(f) 1980s

Figure 4.6: Predicting vintage influence in fashion collections. (a)-(f) indicate the decade of predicted influence.

Fashion influence analysis is a challenging task in which it requires both expertise in the vintage fashion domain and big data since fashion influences or trends are representative of the similar aesthetics that are followed by a group of people, which appear and reappear cyclically over time. Since the deep network demonstrates an excellent performance in capturing the iconic style of an era and is practical within a large dataset, the dating of the historical objects network is proposed as a core module of vintage fashion analysis.

To estimate the influence of vintage fashion on fashion show collections, the fine-tuned model is used to estimate when the collections from the Runway dataset were made. Then, the inspiration date of the collection is defined as the decade with the highest probability among the images from that collection. To evaluate the proposed approach, human judgments on the same task using AMT were collected. For each assignment, five works are shown five fashion show images per collection and ask to identify the decade that inspired these images. Initially 200 collections per predicted decade are picked and



(a) 2000

(b) 2004

(c) 2000-14

Figure 4.7: The influence of vintage fashion (1900s-2000s) in fashion show collections from (a) 2000 and (b) 2004. Figure (c) shows the influence of the fashion of the 1960s, 1980s and 1990s throughout the 2000s - 2010s.

then removed collections with low human agreement (less than 3 of 5 in agreement). The final 300 collections are included for the evaluation. Regarding these collections, the fine-tuned model revealed 58.33% agreement with MAE of 8.6 years compared to the human judgments. Some qualitative results of the temporal prediction task are shown in Figure 4.6.

Once the fashion show collections were dated, it became feasible to analyze the influence of the vintage fashion on modern outfits. To do so, the system accumulates the classification confidence of collections from the same year as shown in Figures 4.7(a) - 4.7(b). By observing the classification confidence of a particular vintage decade across years, several interesting trends are spotted. For example, Figure 4.7(c) shows that the fashion of the 1990s had a strong influence during both the early 2000s and the early 2010s, while during the mid-2000s a revival of the fashion of the 1960s and 1980s fashion occurred.

4.6 Cross-domain fashion influence

Clothing fashion represents an individual's choices about how to represent themselves relative to others. While some choices truly are unique, fashion trends include similar aesthetics that are followed by groups of people which often appear and reappear in a cyclic pattern over time. Seasonal trends like floral patterns in the spring or reds and oranges in the fall recur without fail each year while other fashion trends pop up sporadically, often appearing first on the runway and then filtering quickly into the real world. For example, neon colors reminiscent of the 90s in the Spring 2012 collections from Rodarte, Peter Som, Jason Wu, and Nanette Lepore (among others) have since been appearing in fashion followers' closets.

Thus, this section provides a quantitative analysis of fashion both on the runway and in the real world and further explores the influence of fashion show collections on street outfits.

4.6.1 Outfit similarity

The overarching goals of this section are to propose an automatic system that quantifies and learns to measure outfit similarity both within and across domain scenario and then to use the learned similarity to discover fashion influences or trends.

The subroutine of the proposed approach involves a pair-wise comparison of the clothing outfits based on a learned combination of the visual features that have been aggregated over the semantic parses of clothing. Based on human judgments of outfit similarities, the system trains a discriminative models that replicates human judgments.

Clothing visual representation

To produce the quantitative analyses of fashion, this dissertation presents a novel visual representation that captures the appearance of outfits. To extract the visual representation from an input image, the system starts from pre-processing an image as follows: 1) estimating the pose of a person (Yang and Ramanan, 2011), 2) analyzing what are they wearing in the form of a clothing parse (Yamaguchi et al., 2013), and 3) dividing an image into sub-regions based on the pose and clothing parse.

More specifically, the feature extraction pipeline starts from resizing each person detection window to 320 × 160 pixels. The clothing parsing algorithm is applied to an image in order to identify the clothing pixels or foreground pixels of an image.

Global descriptor: To integrate the global cue to the proposed representation, a Style descriptor (Yamaguchi et al., 2013) is computed over the person detection window. This descriptor is a reduced-dimensional feature of various visual descriptors including RGB, Lab, MR8, Gradients, HOG, Boundary distance and Pose distance.

Local descriptors: With the pose and clothing parse, the system extracts nine sub-regions defined be relevance to the pose estimate as head, chest, torso, left/right arm, hip, left/right/between legs. For each sub-region, four visual features have been extracted from the foreground pixels:

- **Color** Two 512 dimensional histograms in RGB and Lab color spaces from foreground pixels (e.g., clothing items, hair and skin).

- **Texture** A concatenation of two bag-of-words histograms: 1) a histogram of MR8 responses (Varma and Zisserman, 2005) quantized into 256 visual words, and 2) a histogram of HOG descriptors (Dalal and Triggs, 2005) (8x8 blocks, 4 pixel step size, 9 orientations) quantized into 1000 words.

- **Shape** The system starts from resizing each sub-region to 32 × 16 pixels, and extracts 2 features: 1) a binary mask of the foreground pixels estimated by the clothing parsing algorithm, and 2) an edge map of the foreground region estimated using structured forest efficient edge detection (Dollr and Zitnick, 2013). To binarize an edge map using the images parsed as guideline, the threshold t is selected where it minimizes the following cost function:

$$c(t) = \sum_{i \in x} d\left(x_i, \bar{x}_j\right) + \sum_{j \in \bar{x}} d\left(\bar{x}_j, x_i\right) \tag{4.2}$$

  where $d(x_i, \bar{x}_j)$ represents the Euclidean distance transform of the pixel i of binary edge map x thresholded at t to the nearest pixel j of the clothing contour $\bar{x}$. This provides an edge map that has a similar level of detail to the predicted clothing parse boundaries.

- **Parse** An item-masks of 56 different clothing items (e.g., dress, shirt, shoes, etc) which form a 56-dimensional descriptor of the percentage of each item present in the foreground region.

The final representation is a concatenation of the four visual features from nine sub-regions and the Style descriptor.

Human judgments of outfit similarity

Determining *similarities* between images of outfits may be difficult, even for humans. To address this challenge, human judgments of outfit similarity are collected using crowsourcing. Here, agreement between people, where the present provides a definition of the wisdom-of-crowds for similarity between styles.

Human judgment collection: Given a query outfit from the Runway dataset, five annotators on AMT are required to pick the most similar outfit from five candidate outfits, or *none* in the event of no similar outfits being present. The candidate outfits are selected based on the cosine similarity using each individual feature in isolation (e.g., color, texture), or to an equally weighted combination of all features. To explore the outfit similarity under two scenarios, candidate outfits have been taken from both the Runway dataset (within-runway scenario) and the Paper Doll dataset (cross domain scenario). About 2000 human judgments of outfit similarity have been collected for this experiment.

Results: Overall, there is agreement between annotators for the within-runway scenario. For 20.4% of the queries, all five annotators agreed on the most similar outfit. For an additional 29.8% and 24.6% of the queries three and four out of five annotators agree on the best match. In total, the majority of annotators agreed on 74.8% of the queries in the within-runway scenario. The agreement was a little lower in the runway-to-street scenario where all five annotators agreed in 10.9% of the queries, and in 39.3% and 23.7% of the queries, three and four out of five annotators agree on the best match. This has yielded a majority agreement for 73.9% of the runway-to-street queries.

Learning to compare outfits

Using human judgment, the system trains a model of similarity for the fashion outfits. More specifically, a linear SVM (Chang and Lin, 2011) is trained to classify a pair of outfits as either similar or dissimilar. To train the model, three different strategies for converting human judgments to positive/negative labels for training have been explored:

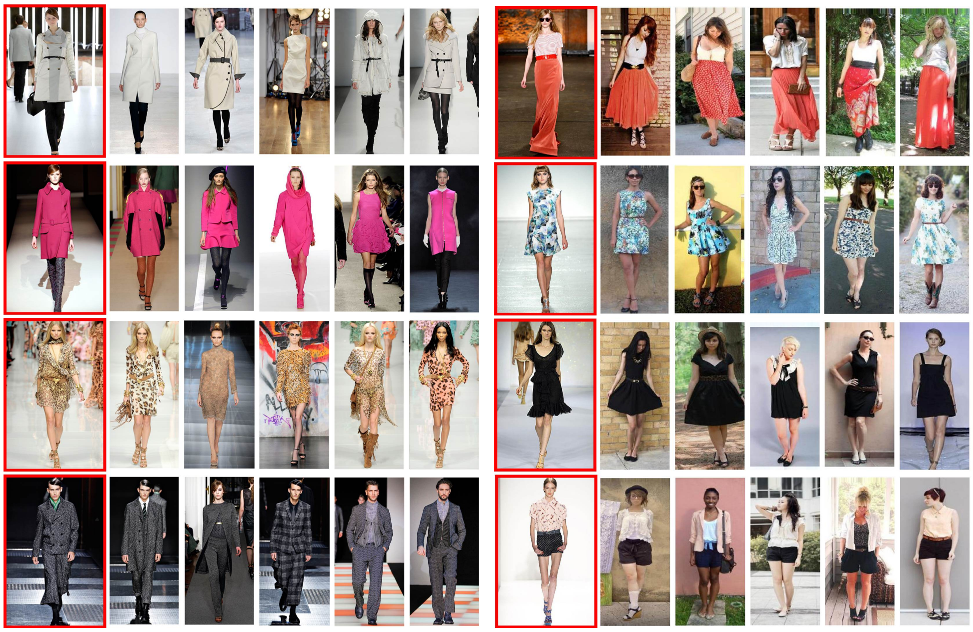
- **Majority:** A query-candidate pair is marked as positive when the pair gets the *majority* of annotators' clicks. Any query for which *all* the annotators clicked none is used to form five negative pairs with each of its five potentially similar images.

- **Unanimity:** Query-candidate pairs for which *all* annotators agree on the best match are treated as positive. Any query for which *all* the annotators clicked none is used to form negative pairs with each of its five potentially similar images.

- **Some:** Query-candidate pairs marked by *any* of the five annotators are treated as positive. And any query for which *all* the annotators clicked none is used to form five negative pairs with each of its five potentially similar images.

Outfit similarity evaluation

Figure 4.9 shows similar outfit retrieval from runway outfit queries. On the left, results reflect the majority approach trained on runway-to-runway labels to retrieve similar runway outfits. On the right results use the majority approach trained on runway-to-realway labels to retrieve realway images from *chictopia.com*. Outfits retrieved both from



(a) Runway to Runway (b) Runway to Realway

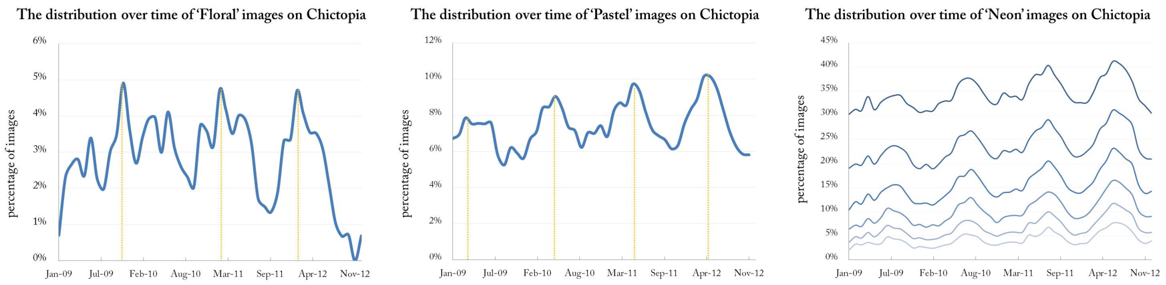
Figure 4.8: Retrieved similar outfits for example query runway outfits (red boxes) using the learned similarity. The right-hand panel shows retrieved outfits from everyday life, while the left-hand panel shows outfits retrieved from other runway collections.

	Runway to Runway		Runway to Realway	
Method	Clothing feature	Style descriptor	Clothing feature	Style descriptor
Majority	0.76 ± 0.11	0.66 ± 0.11	0.54 ± 0.03	0.45 ± 0.02
Unanimity	0.73 ± 0.08	0.62 ± 0.07	0.53 ± 0.02	0.42 ± 0.01
Some	0.73 ± 0.14	0.63 ± 0.12	0.55 ± 0.01	0.43 ± 0.03

Table 4.2: Intrinsic Evaluation: AUC for predicting outfit similarity from Runway images to Runway images or from Runway images to Realway (street-style) images.

the runway and from the realway images look quite promising, colors are matched well, and overall shapes and patterns also tend to be similar.

The quantitative results of the learned similarity model (area under the precision recall curve (AUC) using 6-fold cross validation) for both scenarios are shown in Table 4.2. Establishing the baseline for this experiment involves training the learned similarity model using the Style descriptor (Yamaguchi et al., 2013) as a visual feature. The



(a) Floral print

(b) Pastel colors

(c) Neon colors

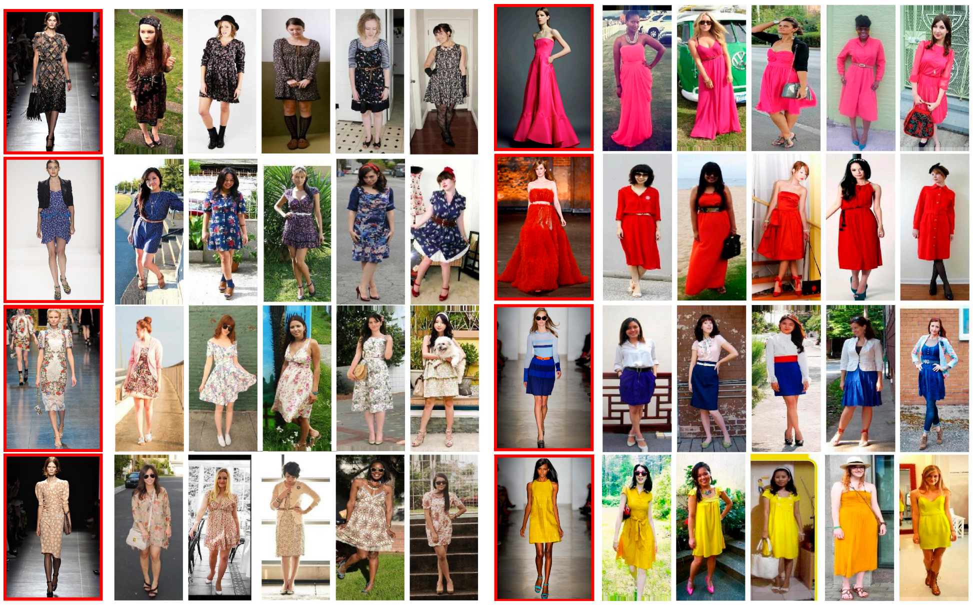
Figure 4.9: Street fashion trends for floral prints, pastel colors and neon colors in the Paper Doll dataset acquired from 2009-2012. The plot shows the density of images similar to the example images for the trends. The number of images is expressed as a fraction of all images posted for that month.

results show that the proposed learned similarity models agree with the human similarity judgments quite well. For the runway-to-runway scenario, learned similarity using the proposed Clothing features and framework achieves $73-76\%$ AUC compared to the Style descriptor baseline of $62-66\%$, giving an increase in performance of about 10% in these experiments. The same trend also appears for the runway-to-realway task ($53-55\%$ vs $42-45\%$).

4.6.2 Influence of runway collections on street fashion

Finally, this section presents the preliminary experiments that examined how runway styles influence street outfits. In particular, the system focuses on images from the runway collections that illustrate three potential visual trends: floral prints, pastel colors and neon colors.

To study these trends, about 110 example images for each trend from the Runway dataset have been manually selected for the experiment. Using each of these runway images as a seed, all street fashion images from the Paper Doll dataset have been retrieved



(a) Floral print retrieval (b) Neon colors retrieval

Figure 4.10: Example retrieval results for floral prints (left) and neon colors (right) trends. Query outfits from the runway are shown in red with retrieved street outfits using the learned similarity

with a similarity score above a fixed threshold, the retrieval results for floral prints and neon colors are shown in Figure 4.10. The percentage of similar images (normalized for increasing dataset size) for each trend has been plotted and is presented in Figure 4.9(a) - 4.9(c). By observing the distribution of the retrieved images over time, the temporal trends at the resolution of months in the street fashion have been spotted. The seasonal variation is clear for all three styles, but neon and pastel colors show a clear increasing trend over time, unlike with the floral style. Moreover, the results shows that even if the similarity threshold is varied, the trend pattern remains the same, as is shown in Figure 4.9(c) where the distribution of the retrieval neon colors images over time with different thresholds are shown.

	AUC		
Methods	Years	Seasons	Brands
Random	0.067	0.333	0.003
Most common	0.151	0.478	0.012
Human	0.240	0.520	-
10-nearest neighbor + f_{style}	0.234	0.534	0.106
10-nearest neighbor + f_{sim}	0.258	0.572	0.122
Classifier + f_{style}	0.244	0.554	0.098
Classifier + f_{sim}	**0.278**	**0.578**	**0.129**

Table 4.3: Extrinsic evaluation: AUC for season, year and brand prediction multi-class classifiers using Style descriptor f_{style} (Yamaguchi et al., 2013) and the proposed feature f_{sim} compared with other baselines.

4.7 Similarity for extrinsic tasks

As fashion similarity may be considered potentially nebulous and subjective, several additional extrinsic tasks have been extended to evaluate the learned similarity.

4.7.1 Predictions

Since, the observation from the Runway dataset shows that clothing from the same season, year or brand shares aspects of a visual appearance (e.g., color, texture, type of fabric, etc.), the season, year and brand prediction problems are proposed as categories of extrinsic evaluation. In this experiment, several prediction frameworks are explored as follows:

- **Random** The system randomizes the predicted season/year/brand of the input image from all of the possibilities (3 seasons, 15 years and 852 brands) with equal probability.

- **Most common** The system identifies the predicted season/year/brand of the input image as the most commonly occurring label.

- **Nearest neighbor** Given the input image, the system retrieves similar images according to the learned similarity f_{sim} (or the Style descriptor f_{style}) and output of the majority classification label. The predicted season, year, or brand is thereby estimated as the majority vote of the top k retrieved runway images from other collections, excluding images from the same collection as the query image. Since the number of possible years and brands is large, sometimes there is no candidate that has a majority vote. In that case, the system randomly predicts the year or brand from the candidate pool.

- **Classifier** The system trains the linear classifier (one vs. all) to predict the year, season, and brand of outfits using the proposed feature f_{sim}, and the Style descriptor f_{style} (Yamaguchi et al., 2013).

To evaluate the difficulty of the prediction tasks, human performance on the prediction tasks are collected using AMT. For season prediction, the definitions and five example images of each season are shown to the annotators at the beginning of the task. Then, five annotators are required to identify the season of an input image. The annotation with the highest agreement among the annotators is then assigned to the input image. Unlike in the season prediction, the annotators are required to choose one of the years without referring to the example images. Notably, due to the large number of distinctive brands in the Runway dataset, human performance on brand prediction is not practical.

Table 4.3 presents a comparison of automatic predictions vs human predictions. The results indicate that humans are better than both the random and most common frameworks. The nearest neighbor techniques based on the learned similarity models performed surprisingly well, and in some cases, even outperformed humans on the same tasks. Moreover, even though the learned similarity is trained for the similarity prediction task, it can achieve a comparable level of performance at about 2% lower than the classifiers that were trained specifically for the extrinsic tasks.

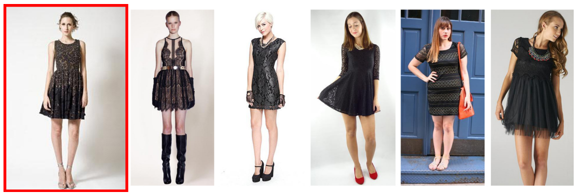
4.7.2 Cross-domain label transfer

In this experiment, the learned similarity has been applied to transfer four different textual tags; style, trend, clothing item and color from the street collections to the runway collections. The label transfer pipeline starts from retrieving the top k-nearest street images of the runway input using the learned similarity. Then, the following alternative approaches are proposed to complete the task.

- **Most similar image** The system directly transfers labels from the most similar realway image to the runway query image.

- **Highest probability** The system retrieves the 10 nearest realway outfits, and selects candidate tags according to their frequency within the set of tags contained in the retrieval set.

- **TFIDF weighting** The system retrieves the 10 nearest realway neighbors, but selects candidate tags weighted according to term frequency (tf) and inverse docu-



(a)

	Most similar	TFIDF	Prob
style	Sexy	Chic	Chic
Trend	Lace dress	CK	Vintage
Item	Dress	Dress	Dress
Color	-	Black	Black

(b)

Figure 4.11: (a) Query runway image(left most), and the 5 nearest realway outfits. (b)The transferred labels from the 10 nearest neighbors of the query image(a)

ment frequency (idf).

Figure 4.11(a) shows an example query image, 5 retrieved realway outfits and tags predicted by each method, while Figure 4.11(b) shows the transferred labels from the 10 nearest neighbors of the query image.

To evaluate the predicted labels, 100 query images are randomly selected and five annotators on the AMT are used to ask to verify whether each label is relevant to the query image. Labels that receive majority agreement (more than three out of five annotators) are counted as relevant. The results show that by using the nearest neighbor technique based on the learned similarity together with TFIDF approach, the system achieved 68%, 30%, 80% and 75% accuracy in this task.

4.8 Conclusion

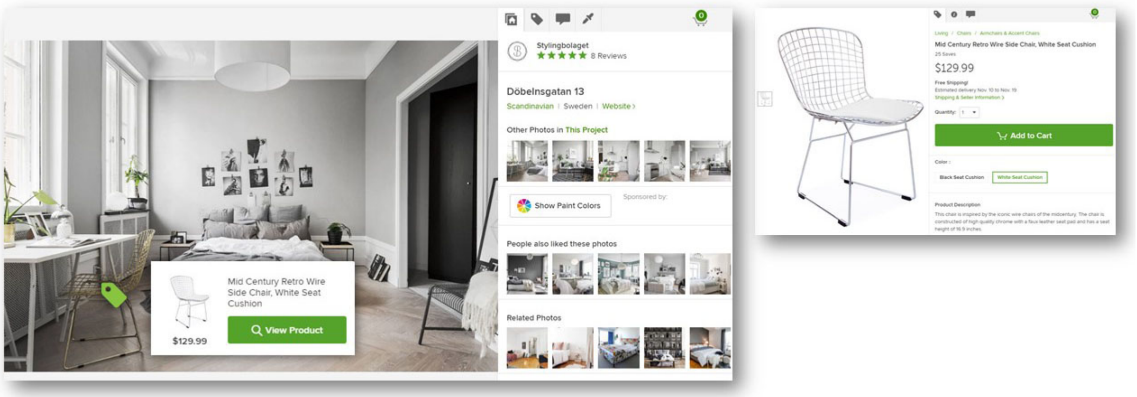
This chapter presents a deep learning approach that would automatically estimate when objects were made, and the evaluation would be done using an existing dataset of cars and two novel datasets of vintage clothing photographs. The neural activations and their entropy would be analyzed in order to gain insights into the temporal sensitivity of the neurons, what the networks have learned and their comparison to the discriminative parts learned by the data mining approach (Lee et al., 2013). While the direct applications of dating historical objects involve large-scale data organization or image retrieval, the results show that the framework can be applied to an analysis of the influence of vintage fashion on runway collections.

To extend the study of the influence of the fashion on runway collections to street fashion, this chapter also presents a novel approach for learning the human judgments of outfit similarity. The results show that the learned similarities match well with human judgments of clothing style similarity in both within-runway and runway-to-street scenario. Finally, the results demonstrate that the learned similarity is practical for many applications: 1) identifying and analyzing the fashion influence or visual trends of runway collections to street fashion, 2) Season, year or brand prediction of runway collections, and 3) Cross-domain label transfer.

CHAPTER 5: CROSS-DOMIAN GENERATIVE ADVERSARIAL NETWORK

This chapter introduces a cross-domain generative adversarial network that has been trained to virtually place a product into a room. There have been numerous attempts at generating realistic images using generative adversarial networks (Goodfellow et al., 2014; Radford et al., 2016; Denton et al., 2015; Dosovitskiy and Brox, 2016a; Nguyen et al., 2016). However, none of the previous works have aimed to take images from two different domains and virtually place one into another in a realistic manner.

Given input images from two different domains: 1) a product image acquired from online shops, and 2) a room image acquired from home design websites, the generative adversarial network is trained to generate a realistic image of the input product in the room. The network is trained to automatically transform (i.e., translate, rotate or scale) a product image; then, generate a realistic image of the product in a room while the visual attributes (e.g., color, texture, shape, etc.) of the product are preserved. The proposed generative adversarial network simultaneously trains two models: 1) a generator that captures the data distributions from both domains, and 2) a discriminator that estimates the probability that an input image came from the training data rather than appeared as synthesized data from the generator.


Mid Century Retro Wire Side Chair, White Seat Cushion
$129.99
View Product
Stylingbolaget
★★★★★ 8 Reviews
Döbelnsgatan 13
Scandinavian | Sweden | Website ›
Other Photos in This Project
Show Paint Colors
Sponsored by:
People also liked these photos
Related Photos
Living / Chairs / Armchairs & Accent Chairs
Mid Century Retro Wire Side Chair, White Seat Cushion
25 Saves
$129.99
Free Shipping!
Estimated delivery Nov. 10 to Nov. 19
Shipping & Seller Information ›
Quantity: 1
Add to Cart
Color :
Black Seat Cushion
White Seat Cushion
Product Description

This chair is inspired by the iconic wire chairs of the midcentury. The chair is constructed of high quality chrome with a faux leather seat pad and has a seat height of 16.9 inches.

Figure 5.1: Example of the bedroom image from the Houzz dataset on the right and the corresponding product image from the Product dataset on the left. The green tag on the bottom left of the room image indicates the pinpoint location of the chair which is linked to the product page on the online shop.

5.1 Datasets

This chapter introduces two novel datasets from two different domains: 1) a collection of indoor and outdoor scenes designed by professional interior designers, and 2) a collection of shop photographs of furniture and home decoration products.

5.1.1 Houzz dataset

Houzz dataset is a collection of photographs of the interiors and the exteriors of home designs acquired from *houzz.com*; an online platform for home designers and home remodeling professionals around the world that is used as a medium of conveyance to showcase their work, as well as being a place to collaborate with clients and/or prospective clients.

The dataset consists of 136,034 room listings across 20 styles (e.g., contemporary, modern, traditional, etc.) and 30 categories (e.g., bedroom, living room, garage, etc.).

Each room listing from the Houzz dataset contains an image, a designer's name, a title, room description, room category, style, location, number of saved ideabooks, Q&A, comments, keywords, links of products that appear in the image and their associated pinpoint locations as shown in Figure 5.1.

5.1.2 Product dataset

Product dataset contains 83,706 shop photographs of 720 product categories (e.g., chandeliers, coffee tables, sofa, etc.) over 19 styles (e.g., Asian, beach style, Victorian, etc.) acquired from *houzz.com*. Each product listing consists of a product image, vendor name, title, price, product description, category, styles, number of saved ideabooks, Q&A, comments, keywords and links to the rooms where the product appears. The relationship between the room images and the product images are many-to-many: one product could appear in multiple rooms and one room could contain multiple products.

5.2 Product localization

To provide the most satisfying service to the clients, the designers not only provide the most stylish room images but also the sources of the products in the rooms. In each room from the Houzz dataset, many products are linked to their corresponding product listing in the online shops via the pinpoint location in the image of the room. Although the product links and the pinpoint location of a product in the image are provided by the interior designers themselves, the data can be noisy:

- **Inaccurate location** Many of the pinpoint locations are at the edge of the product region. And in the worst case, they are not inside the product region.

- **Inconsistent product link** Since the shopping websites occasionally update their products, some product links associate to similar products (i.e., same products but they are different in color or texture), irrelevant products or out of stock products where the associated product data no longer exist.

- **Visual appearance inconsistent** The major characteristics of the Product dataset include the following: 1) frontal view product image with no occlusion, 2) a single product appears in the center of an image, and 3) the background pixels are plain in color (e.g., white, black, etc.). However, the associated room images from the Houzz dataset are more complex. There are multiple products in a room and they are carefully placed together to create the most functional and beautiful space which can lead to many issues: 1) many products appear from a different view point from the product image, 2) some products are very small in the room, and 3) there is a great deal of occlusion of products in rooms by other objects.

Since the characteristics of the Product dataset and the Houzz dataset are different, two different product localization frameworks have been designed to propose the product bounding boxes for each dataset.

Product dataset localization: To localize the product location in the product image I_P from a product image domain $P \subset \mathbb{R}^{W \times H \times 3}$ from the Product dataset, the system applies the following procedure:

1. Compute the average background color A_i of 4 background patches of a size of 3×3 from the corners of a product image I_{P_i} of product i.

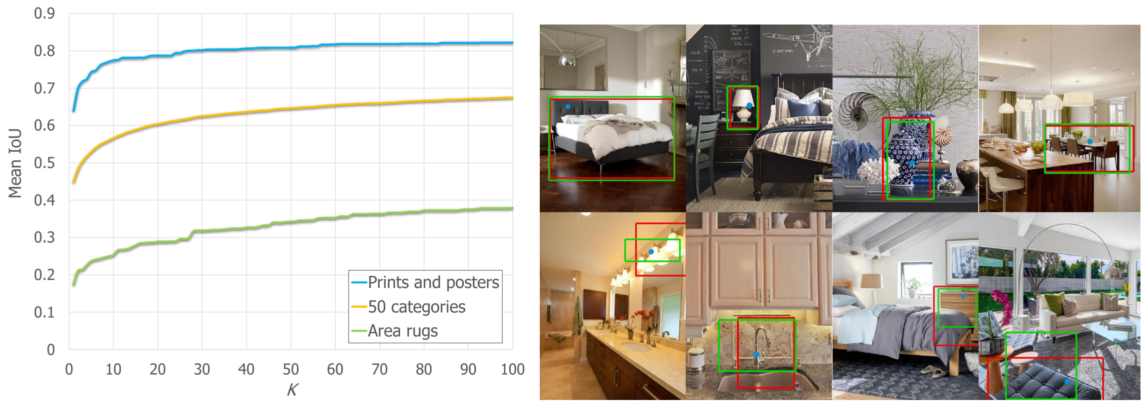
2. Calculate the ‘objectness’ of the product $O_i = |A_i - I_{P_i}|$.
3. Classify the image pixels $I_{P_i}(x, y)$ into foreground pixels where $O_i(x, y) \geq \theta$.
4. Finally, compute the product bounding box B_i as the tight bounding box over foreground pixels. The product patch $I_{P_i}^B \in I_{P_i}$ is then defined as the product region inside B_i.

Houzz dataset localization: Given the product image I_{P_i} from the Product dataset and the associated room image I_{R_j} from the room image domain $R \subset \mathbb{R}^{W\times H\times 3}$, the system applies the following procedure:

1. Sample the sliding-windows of M multi-scale image patches $p_m^j \in I_{R_j}, m \in M$ from the input image from the Houzz dataset with a stride size of 10. In the experiments, the patch sizes range from $5 - 80\%$ of the room image size with the same aspect ratio with the product bounding box B_i. The image patches which do not contain the pinpoint location of the product will be discarded.
2. Represent the image patches p_m^j and product patch $I_{P_i}^B$ with the neuron activations from the last fully-connected layer (FC7) of VGG-16 (Simonyan and Zisserman, 2014).
3. Calculate the Euclidean distance between p_m^j and $I_{P_i}^B$ in the feature domain, the bounding box with the smallest L2 distance is then defined as the product bounding box B_i^j of the product i in the room j.

5.2.1 Product localization evaluation

To evaluate the product localization framework, the ground truth bounding boxes are collected from the Amazon Mechanical Turk (AMT). The fifty product categories with



(a) Mean IoU

(b) Product bounding box results

Figure 5.2: The product localization performance: (a) mean IoU of top K product bounding boxes from the room images. The plot shows the average IoU score over 50 product categories, the Prints and posters category and the Area rugs category, and (b) the examples of the product bounding boxes acquired from different categories in green compared with ground truth bounding boxes in red and the pinpoint locations in blue points.

forty associated room images per category are randomized for the evaluation purpose. To collect the ground truth bounding box, three annotators are required to draw a bounding box of a given product in a given room image.

Figure 5.2(a) shows the mean intersection-over-union (IoU) score of the ground truth bounding boxes and the top K proposed bounding boxes of the product in the room image at different K values. At each K, the graph shows the highest IoU score of the ground truth bounding boxes and K proposed bounding boxes, averaging across images. The results show that for *Prints and photos* category gained the higher IoU score than the *Area rugs* category. One reason is that the variation between the print images from the product domain and the room domain were smaller than the area rug images. The area rugs usually appear on the floor of the room which is not only presented from a different viewpoint from the product image (frontal view), but the area rugs are also usually occluded by other products and, the prints or photos usually appear on the wall and present less viewpoint variations and less occlusion.

Figure 5.2(b) shows the examples of the ground truth product bounding box in a red rectangle, the proposed product bounding box in a green rectangle and the blue points shows the pinpoint location of the product. While the first row shows the successful examples of the product localization framework, the bottom row shows the failure examples. The results show that the proposed product localization framework tends to fail when the algorithm fails to estimate the aspect ratio of the product. Moreover, the framework also suffers from occlusion and viewpoint variations. Finally, the results also show that the product pinpoint locations are noisy, as 77.5% of the pinpoint locations from the Houzz dataset are inside the ground truth bounding boxes.

5.3 Cross-domain image generation

This section introduces the cross-domain image generation problem. Given input images from two different domains: 1) a product image acquired from online shops, and 2) a room image acquired from home design websites, the generative adversarial network is trained to generate a realistic image of the input product in the room. The network is trained to automatically transform (i.e., translate, rotate or scale) a product image, then to generate a realistic image of a product in a room while the visual attributes (e.g., color, texture, shape, etc.) of the product are preserved.

5.3.1 Generator network

Given a transfer function, generator G, the task is to transfer both a product image $I_P \in P$ and a room image $I_R \in R$ to the new room image $\hat{I_R} \in R$ such as

$$\hat{I_R} = G(I_P, I_R|\Theta^G) \tag{5.1}$$

where Θ^G is the model parameter of the generator. In the experiments, the system applies a convolutional network model for generator G.

Generator G is employed as a standard encoder-decoder pipeline. The encoder takes two input images: 1) a product image I_P of size 64×64, and 2) a room image I_R of size 128×128 with a bounding box B of size 64×64 which determines the specific the location of a product in the final image. From the inputs, the encoder produces a latent feature representation which captures the structure of the product and the room. The decoder takes this feature representation and transfers the product into the room content inside the bounding box. In this experiment, two different encoder architectures are explored.

Single-path encoder Given two input images: 1) a product image I_P of size 64×64, and 2) a room image I_R of size 128×128 with a bounding box B of size 64×64 which determines the specific location of a product in the synthesized image $\hat{I_R}$. The system begins by replacing the room pixels inside the bounding box with product pixels, resulting in a single input $\tilde{I_R}$. Then, $\tilde{I_R}$ is forwarded to the encoder with a series of five convolutional layers with a kernel size of 4×4 and a stride of 2. Each convolutional layer, except the last layer, is followed by a batch normalization operation (Ioffe and Szegedy,

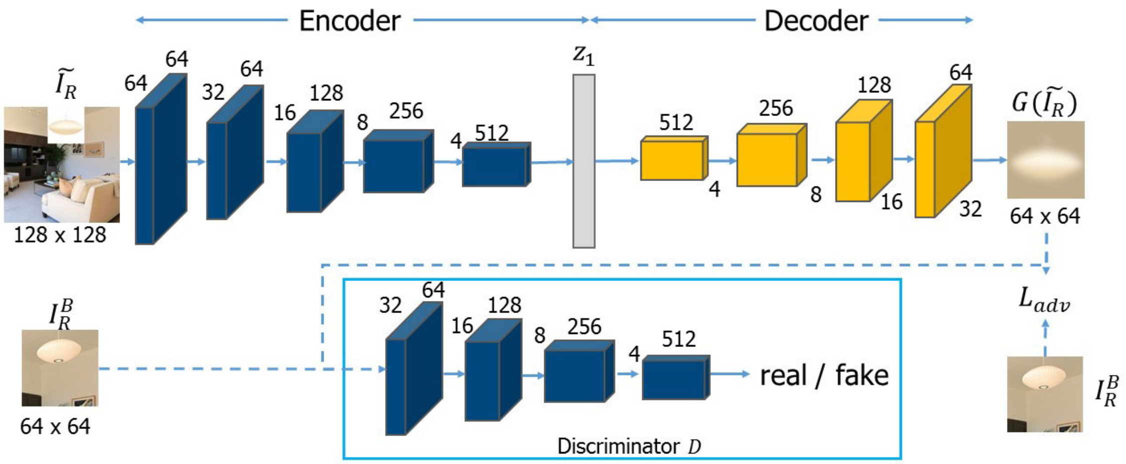


Figure 5.3: The adversarial network architecture with the single-path encoder.

2015) and LeakyReLU activation function (Nair and Hinton, 2010). The final latent feature representation z_1 of size 4096 is then fed to the decoder. The overall pipeline of the network with a single-path encoder is shown in Figure 5.3.

Two-path encoder The generator G is derived from the Siamese architecture as shown in Figure 5.4. The encoder consists of two asymmetrical paths of convolutional layers: 1) the product image path for a product image, and 2) the room image path for a room image. The product image path consists of five convolutional layers with a kernel size of 4×4 and a stride of 2. Each convolutional layer, except the last layer, is followed by a batch normalization operation (Ioffe and Szegedy, 2015) and LeakyReLU activation function (Nair and Hinton, 2010). The size of the latent feature representation of the product image path z_1 is 512. Since the input room image is two times larger than the product image, the additional convolutional layer followed by a batch normalization operation and LeakyReLU activation function are added into the room image path. And the size of the latent feature representation of the room image stack z_2 is 3,584. The

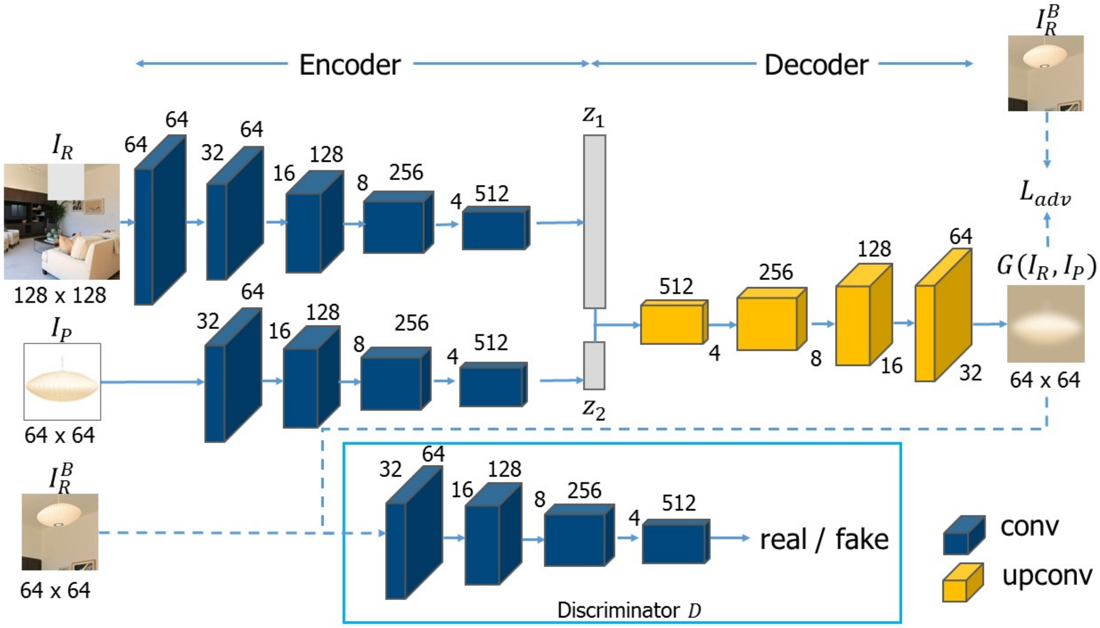


Figure 5.4: The adversarial network architecture with the two-path encoder

two paths are sequential and trained independently without shared weights. Finally, the hidden variables z_1 and z_2 acquired from both paths are concatenated and fed into the decoder.

Decoder The decoder consists of a series of five up-convolutional layers (A.Dosovitskiy et al., 2015) with learned filters, each with a batch normalization operation and a rectified linear unit (ReLU) activation function (except the last up-convolutional layer). An up-convolutional is an upsampling operation followed by the convolution that results in a higher resolution image. Finally, the Tanh function is applied as the last layer of the decoder.

5.3.2 Discriminator networks

The discriminator D takes either a room image I_R^B of size 64×64 (an image patch inside the bounding box) or a synthesized image $\hat{I_R}$ from generator G, and distinguishes whether its input is real or fake (i.e., synthesized image). Generator G and discriminator D are adversaries since G is trained to estimate the data distribution or maximize the probability of D making a mistake.

The real/fake discriminator D architecture is similar to that of the generator (the product image path). The discriminator consists of five convolutional layers with a kernel size of 4×4 and a stride of 2. Each convolutional layer, except the last layer, is followed by a batch normalization operation and LeakyReLU activation function. The last layer of D is a Sigmoid layer where its output is the real number which will be large when the input comes from the training set, and will be small when the input image is the synthesized image from G.

5.3.3 Loss function

For a pair, such as product image I_P and room image I_R with the bounding box B, the adversarial network is trained by regressing to the ground truth content inside the bounding box I_R^B. Let M_R be a binary mask corresponding to the product bounding box B in the room image I_R with a value of 1 wherever the pixel is inside the bounding box and 0 for otherwise. During the training, the masks are automatically generated for each image and training iterations. The components of the loss function of the model are described in the following section.

Generator loss: The generator loss $\mathcal{L}_{gen}$ is responsible for capturing the overall structure of the region inside the bounding box and for a value of coherence with regard to its context. To achieve this goal, the generator loss $\mathcal{L}_{gen}$ is defined as the normalize masked $L2$ distance:

$$\begin{aligned}\mathcal{L}_{gen}(\tilde{I}_R) &= \left\| M_R \odot (I_R - G(\tilde{I}_R)) \right\|_2^2, \\ \mathcal{L}_{gen}(I_R, I_P) &= \| M_R \odot (I_R - G((1 - M_R) \odot I_R, I_P)) \|_2^2\end{aligned} \tag{5.2}$$

where $\mathcal{L}_{gen}(\tilde{I}_R), \mathcal{L}_{gen}(I_R, I_P)$ is defined for the single-path encoder and the two-path encoder, respectively, and $\odot$ is the element-wise product operation.

Discriminator loss: Unlike the generator, the objective for the discriminator is the logistic likelihood indicating whether the input is real or synthesized:

$$\min_G \max_D \mathbb{E}_{I_R \in R} \left[log(D(I_R)) \right] + \mathbb{E}_{I_R \in R, I_P \in P} \left[log(1 - D(G(I_R, I_P))) \right] \tag{5.3}$$

This method has recently shown promising results in many image generation studies (Goodfellow et al., 2014; Radford et al., 2016). To condition the given context information, the adversarial loss $\mathcal{L}_{adv}$ is defined as:

$$\mathcal{L}_{adv} = \max_D \mathbb{E}_{I_R \in R, I_P \in P} \left[log(D(I_R)) + log(1 - D(G((1 - M_R) \odot I_R, I_P))) \right] \tag{5.4}$$

where, in practice, both D and G are optimized jointly using alternating SGD. Finally, the overall loss function is defined as:

$$\mathcal{L} = \lambda_{gen}\mathcal{L}_{gen} + \lambda_{adv}\mathcal{L}_{adv} \tag{5.5}$$

where $\lambda_{gen}, \lambda_{adv}$ are the weight parameters.

5.4 Experiments

This section begins with the implementation details for the experiments. Then, the section discusses the dataset pre-processing and augmentation, and the quantitative results of both architectures: 1) the adversarial network with a single-path encoder, and 2) the adversarial network with a two-path encoder.

5.4.1 Implementation details

Unless noted, the following training process and parameter setting details are applied to all experiments. The overall loss function is jointly trained with the weight $\lambda_{gen} = 0.999$ and $\lambda_{adv} = 0.001$. The networks are implemented in Torch and utilize the ADAM stochastic gradient descent solver (Kingma and Ba, 2014) with a mini-batch size of 64 for optimization. To further emphasize the consistency of prediction within the context, the network applies a higher learning rate for the generator ($\times 10$ times) than the discriminator ($lr = 0.0002$).

5.4.2 Data pre-processing and augmentation

The system begins by splitting the room images from the Houzz dataset and their associated product images from the Product dataset at a ratio of 0.85:0.15 for the training and testing sets.

Given two input images: 1) room images from the Houzz dataset, and 2) product image from the Product dataset, the system localizes the product bounding box B of the associate product in the room (Sec. 5.2). Note that B indicates the target region of the product in the room which can also be specified by the user during the testing. The proposed bounding boxes B are then resized to a size of 64×64, and their associated room images are resized with the same aspect ratio. The region of the size 128×128 of the room image around B is then cropped, resulting in the room image I_R for the training. The product image I_P is simply resized to a size of 64×64 for the training as well. To prevent the network from over-fitting, the system performs data augmentation on training the room-product pairs by using the top 3 proposed bounding boxes to generate I_R.

To prepare a single input image for the single-path encoder pipeline, the system directly replaces the pixels inside B with I_P, resulting in a single input image $\tilde{I}_R$ for the generator. For the two-path encoder pipeline, the system fills the pixels inside B with the constant mean value from the training set.

5.4.3 Experimental results

Figure 5.5 shows the qualitative results from two different adversarial networks: 1) Single-path generative adversarial network, and 2) Two-path generative adversarial network. During the training, each model was trained on two different training sets: 1) the training set was randomized from all product categories resulting in about 300K training pairs, and 2) the training set that was randomized from 'Chandeliers' category resulting in about 8K training pairs. For all-category models, the models were trained for 5 epochs, while the chandelier models were trained for 500 epochs.

Figure 5.5(b) shows six pairs of input-output images from the single-path generative adversarial network. Given the input images $\tilde{I}_R$ on the left, the synthesized images $\hat{I_R}$ are shown on the right. Figure 5.5(b) shows four triplets of room-product-output images from the two-path generative adversarial network. Given two input images: 1) the room images I_R with the product bounding box B (filled with the constant mean value) on the left most, and 2) the product images I_P in the middle, along with the synthesized images $\hat{I_R}$ being shown on the right.

The results from the all-category model (rows 1-6) and the chandelier model (rows 7-8), acquired from both Figures 5.5(a) - 5.5(b) show that the models had difficulty in transferring the product pixels from I_P to $\hat{I_R}$. Instead of transferring pixels from one domain to another, the models take a shortcut solution by generating the contents of the missing pixels conditioned on the image surroundings or impainting due to the following explanations.

Firstly, $\mathcal{L}_{gen}$ as the normalized masked $L2$ distance is not an efficient loss for cross-domain image generation. With $L2$ loss, the generator can ignore the product pixels by generating the low-level structure of the room image (e.g., the continuing boundary patterns or edges/textures) and achieves a reasonable loss.

Secondly, the cross-domain image generation is more complicated than the impainting task. In order to transfer the product pixels from one domain to another in a realistic manner, G is required to learn various pixel transformations (i.e., rotation, scaling, view point transformation, etc.) with different degrees. Although training a single network to solve the problem might not be the optimal solution, training a network for a transformation is not practiced due to the limitations of the training samples.

5.5 Discussion

To enforce the network to be able to transfer product pixels from one domain to another without taking a shortcut (e.g., impainting), several techniques are employed and discussed as follows:

- **Weighted Euclidean distance** To emphasize the importance of product pixels, the Gaussian weight w is introduced to the generator loss $\mathcal{L}_{gen}$:

  $$\begin{aligned}\mathcal{L}_{gen}(\tilde{I}_R) =&\lambda \left\| M_R \odot (w \odot (I_P - G(\tilde{I}_R))) \right\|_2^2 \\ &+ (1-\lambda) \left\| M_R \odot ((1-w) \odot (I_R - G(\tilde{I}_R))) \right\|_2^2\end{aligned} \tag{5.6}$$

  where λ is the weight of the loss. In this scenario, G is enforced to generate an image where its center pixels are similar to the product image and the background pixels are similar to the room image.

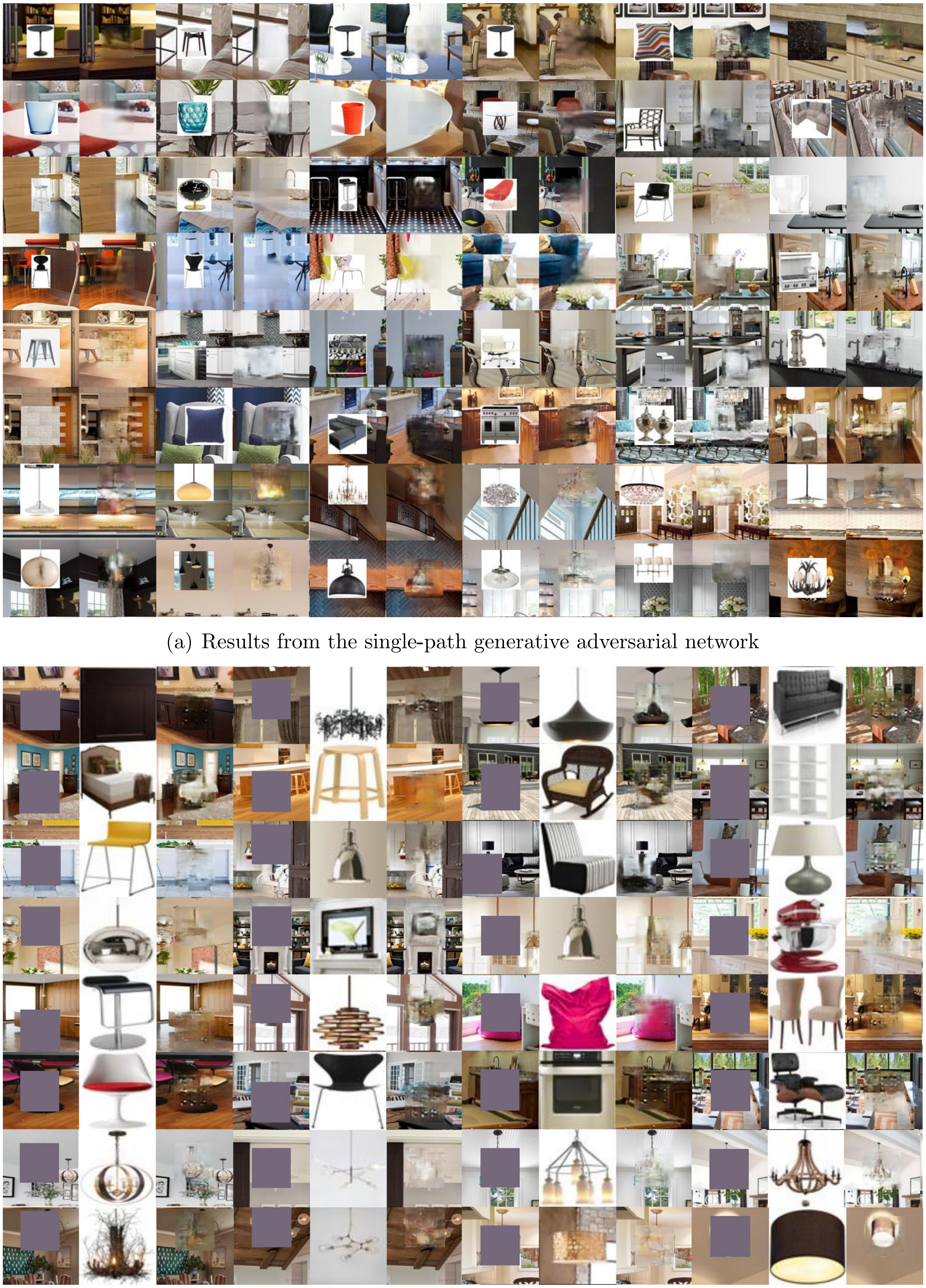
- **Domain discriminator** The domain discriminator D_A (Yoo et al., 2016) takes a pair of images from two different domains and produces a scalar probability of whether or not the input pair is associated. Given the product image I_P, the ground truth content I_R^B, the synthesize image $\hat{I}_R = G(I_R, I_P)$ and the random room image I'_R, the domain discriminator loss $\mathcal{L}_{adv}^{D_A}$ is defined as:

$$\mathcal{L}_{adv}^{D_A}(I_P, I) = -t \cdot log\left[D_A(I_P, I)\right] + (t-1) \cdot log\left[1 - D_A(I_P, I)\right],$$
$$\text{s.t., } t = \begin{cases} 1 & \text{if} \quad I = I_R^B \\ 0 & \text{if} \quad I = \hat{I}_R, I'_R \end{cases} \tag{5.7}$$

The product image I_P is always one of the input pairs while the other I is randomized among $(I_R^B, \hat{I}_R, I'_R)$ with equal probability. The domain discriminator is trained to produce high probability when the input pair is (I_P, I_R^B), otherwise the probability should be low.

5.6 Conclusion

This chapter proposes the cross-domain image generation problem. Given two input images from two different domains: 1) shopping image, and 2) scene image, this chapter explores generative adversarial networks for transferring the product from the shopping image to the scene image such that: 1) the output image looks realistic, and 2) the visual attributes of the product are preserved. Although the generative adversarial networks demonstrated convincing results for several related problems, the results have shown that there is room to improve the system in order to address this problem. Several techniques have been discussed in this chapter and will be integrated in future work.



(a) Results from the single-path generative adversarial network

(b) Results from the two-path generative adversarial network

Figure 5.5: Qualitative results from two generative adversarial networks: a) Single-path generative adversarial network, and b) Two-path generative adversarial network.

CHAPTER 6: CONCLUSION

This dissertation exploited weakly-supervised data from the Internet in order to construct computational methodologies so as to: 1) discover and recognize visual attributes of objects, 2) estimate the temporal attributes of objects, and 3) synthesize novel images of objects while preserving their visual attributes.

First, this dissertation utilized the convolutional neural network for visual attribute discovery and recognition. This dissertation is the first attempt to explore the visual attributable words in terms of perception inside deep neural networks. The discovery pipeline of this study focused on the analysis of neural activations in order to identify the degree of being visually perceptible, namely the *visualness* of a given attribute. The principle concept of this process involves the use of neurons that are highly activated in the network, and that are identified by the KL divergence of their activation distribution in a weakly annotated dataset. The empirical study using two novel datasets demonstrated that the system automatically learned a visual attribute classifier that has a similar perceptual ability to humans. The extended experimental results showed that the neural activations are also practical for the visual attribute localization task. Moreover, the system also exploited an advantage of the layered structure of the deep model to determine the semantic depth of the attribute. And the results revealed that the depth of the deep network corresponded to the depth of attribute perception.

Beyond discovering the simple visual attributes (e.g., color, texture, etc.), this dissertation examined certain more informative attributes, particularly the temporal attributes since they can lead to certain other relevant and constructive applications, such as those involved with trend analysis and prediction problems. The dissertation proposed a deep learning approach that automatically dated when objects were made. The dating historical object networks were evaluated using both an existing dataset and two novel datasets. The results demonstrated the state-of-the-art performance of all datasets. Moreover, the neural activations and their entropy were analyzed in order to provide insights into the temporal sensitivity of the neurons, what the networks have learned and their comparison to the discriminative parts learned by the data mining approach. The insights obtained from the experiments are useful for improving the dating historical object network in the future. To further utilize the dating historical object network, this dissertation applied the dating framework to analyze the influence of vintage fashion on runway collections as well as the influence of fashion on runway collections and on street fashion. The results indicated that the proposed approach could discover certain fashion trends of street fashion, inspiring many applications such as fashion trend predictions and fashion recommendation or advertisements based on current fashion trends.

Finally, this dissertation employed the generative adversarial network for the realistic image generation task. Given two input images from two different domains: 1) shopping image, and 2) scene image, the dissertation explored the generative models for transferring the product from the shopping image to the scene image such that: 1) the output image looks realistic and 2) the visual attributes of the product are preserved. The dissertation reviewed two baselines, and both demonstrated the promising results. To improve the baseline performances, many state-of-the-art image generation techniques have been discussed for use in future work. Although this dissertation only included the experiments in the early state of the study, the cross-domain image generation demonstrated its potential in a range of other applications. For example, the method can be applied to: 1) a home decoration application which gives the users the power to experiment with home decor options by virtually placing products from the online shops into their homes before they decide to buy them, or 2) an online clothing website which provides a new shopping experience wherein the users can virtually try on the clothes (full-body shots) from the comfort of their homes before purchasing them.

BIBLOGRAPHY

A.Dosovitskiy, J.T.Springenberg, and T.Brox (2015). Learning to generate chairs with convolutional neural networks. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Berg, T. L., Berg, A. C., and Shih, J. (2010). Automatic attribute discovery and characterization from noisy web data. In *The European Conference on Computer Vision (ECCV)*, pages 663–676. Springer.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.

Chen, H., Gallagher, A., and Girod, B. (2012). Describing clothing by semantic attributes. In *The European Conference on Computer Vision (ECCV)*, pages 609–623.

Chen, X., Shrivastava, A., and Gupta, A. (2013). Neil: Extracting visual knowledge from web data. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 1409–1416.

Cushen, G. A. and Nixon, M. S. (2013). Mobile visual clothing search. In *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, pages 1–6. IEEE.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893 vol. 1.

de Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *The International Conference on Language Resources and Evaluation(LREC)*, volume 6, pages 449–454.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Denton, E. L., Chintala, S., Fergus, R., et al. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494.

Divvala, S., Farhadi, A., and Guestrin, C. (2014). Learning everything about anything: Webly-supervised visual concept learning. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Doersch, C., Gupta, A., and Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *The IEEE International Conference on Computer Vision (ICCV)*.

Doersch, C., Singh, S., Gupta, A., Sivic, J., and Efros, A. (2012). What makes paris look like paris? *ACM ToG*, 31(4).

Dollr, P. and Zitnick, C. L. (2013). Structured forests for fast edge detection. In *The IEEE International Conference on Computer Vision (ICCV)*, ICCV '13, pages 1841–1848, Washington, DC, USA. IEEE Computer Society.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2013). Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*.

Dong, J., Chen, Q., Xia, W., Huang, Z., and Yan, S. (2013). A deformable mixture parsing model with parselets. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 3408–3415.

Dosovitskiy, A. and Brox, T. (2016a). Generating images with perceptual similarity metrics based on deep networks. *arXiv preprint arXiv:1602.02644*.

Dosovitskiy, A. and Brox, T. (2016b). Inverting visual representations with convolutional networks. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Duan, K., Parikh, D., Crandall, D., and Grauman, K. (2012). Discovering localized attributes for fine-grained recognition. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3474–3481. IEEE.

Escorcia, V., Niebles, J. C., and Ghanem, B. (2015). On the relationship between visual attributes and convolutional networks. *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1256–1264.

Everingham, M., Van Gool, L., Williams, C., Winn, J., and Zisserman, A. (2008). The pascal visual object classes challenge 2008 (voc 2008) results (2008).

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. (2009). Describing objects by their attributes. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1778–1785.

Ferrari, V. and Zisserman, A. (2007). Learning visual attributes. In *Advances in Neural Information Processing Systems*, pages 433–440.

Fischer, P., Dosovitskiy, A., and Brox, T. (2014). Descriptor matching with convolutional neural networks: a comparison to sift. *arXiv preprint arXiv:1405.5769*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587. IEEE.

Gkioxari, G., Hariharan, B., Girshick, R., and Malik, J. (2014). R-cnns for pose estimation and action detection.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., and Wierstra, D. (2015). Draw: A recurrent neural network for image generation. *The International Conference on Machine Learning (ICML)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv*, 7(3):171–180.

Huang, C., Change Loy, C., and Tang, X. (2016). Unsupervised learning of discriminative attributes and visual representations. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv*.

Kalantidis, Y., Kennedy, L., and Li, L.-J. (2013). Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 105–112. ACM.

Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., and Winnemoeller, H. (2014). Recognizing image style. In *The British Machine Vision Conference (BMVC)*.

Kiapour, M. H., Han, X., Lazebnik, S., Berg, A. C., and Berg, T. L. (2015). Where to Buy It: Matching Street Clothing Photos in Online Shops. *The IEEE International Conference on Computer Vision (ICCV)*, pages 3343–3351.

Kiapour, M. H., Yamaguchi, K., Berg, A. C., and Berg, T. L. (2014). Hipster wars: Discovering elements of fashion styles. *The European Conference on Computer Vision (ECCV)*, pages 472–488.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kovashka, A., Parikh, D., and Grauman, K. (2012). Whittlesearch: Image search with relative attribute feedback. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2973–2980. IEEE.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., Bernstein, M., and Fei-Fei, L. (2016). Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *The Neural Information Processing Systems (NIPS)*.

Kumar, N., Berg, A., Belhumeur, P. N., and Nayar, S. (2011). Describable visual attributes for face verification and image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1962–1977.

Kumar, N., Berg, A. C., Belhumeur, P. N., and Nayar, S. K. (2009). Attribute and simile classifiers for face verification. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 365–372.

Kwak, I. S., Murillo, A. C., Belhumeur, P. N., Kriegman, D., and Belongie, S. (2013). From bikers to surfers: Visual recognition of urban tribes.

Lampert, C. H., Nickisch, H., and Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 951–958. IEEE.

Lampert, C. H., Nickisch, H., and Harmeling, S. (2014). Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465.

Lee, Y. J., Efros, A. A., and Hebert, M. (2013). Style-aware mid-level representation for discovering visual connections in space and time. In *The IEEE International Conference on Computer Vision (ICCV)*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *The European Conference on Computer Vision (ECCV)*, pages 740–755. Springer.

Liu, S., Feng, J., Domokos, C., Xu, H., Huang, J., Hu, Z., and Yan, S. (2014). Fashion parsing with weak color-category labels. *Multimedia, IEEE Transactions on*, 16(1):253–265.

Liu, S., Feng, J., Song, Z., Zhang, T., Lu, H., Xu, C., and Yan, S. (2012a). Hi, magic closet, tell me what to wear! In *ACM international conference on Multimedia*, pages 619–628. ACM.

Liu, S., Song, Z., Liu, G., Xu, C., Lu, H., and Yan, S. (2012b). Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3330–3337.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *The IEEE International Conference on Computer Vision (ICCV)*.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440.

Long, J. L., Zhang, N., and Darrell, T. (2014). Do convnets learn correspondence? In *The Neural Information Processing Systems (NIPS)*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*.

Mahendran, A. and Vedaldi, A. (2015). Understanding deep image representations by inverting them. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5188–5196. IEEE.

Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Team, T. G. B., Pickett, J. P., Holberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M. A., and Aiden, E. L. (2010). Quantitative analysis of culture using millions of digitized books. *Science*.

Murillo, A. C., Kwak, I. S., Bourdev, L., Kriegman, D., and Belongie, S. (2012). Urban tribes: Analyzing group photos from a social perspective. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 28–35. IEEE.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *The International Conference on Machine Learning (ICML)*, pages 807–814.

Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., and Clune, J. (2016). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *arXiv preprint arXiv:1605.09304*.

Ozeki, M. and Okatani, T. (2014). Understanding convolutional neural networks in terms of category-level attributes. In *The Asian Conference on Computer Vision (ACCV)*, pages 362–375. Springer.

Palermo, F., Hays, J., and Efros, A. A. (2012). Dating historical color images. In *The European Conference on Computer Vision (ECCV)*, pages 499–512.

Parikh, D. and Grauman, K. (2011). Relative attributes. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 503–510. IEEE.

Patterson, G. and Hays, J. (2012). Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Patterson, G. and Hays, J. (2016). *COCO Attributes: Attributes for People, Animals, and Objects*, pages 85–100. Springer International Publishing.

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks.

Raptis, M., Kokkinos, I., and Soatto, S. (2012). Discovering discriminative action parts from mid-level video representations. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Rastegari, M., Farhadi, A., and Forsyth, D. (2012). Attribute discovery via predictable discriminative binary codes. In *The European Conference on Computer Vision (ECCV).*

Sermanet, P., Kavukcuoglu, K., Chintala, S., and LeCun, Y. (2013). Pedestrian detection with unsupervised multi-stage feature learning. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR).*

Shankar, S., Garg, V. K., and Cipolla, R. (2015). Deep-carving: Discovering visual attributes by carving deep neural nets. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3403–3412.

Shao, M., Li, L., and Fu, Y. (2013). What do you do? occupation recognition in a photo via social context. *The IEEE International Conference on Computer Vision (ICCV).*

Shrivastava, A., Singh, S., and Gupta, A. (2012). Constrained semi-supervised learning using attributes and comparative attributes. In *The European Conference on Computer Vision (ECCV)*, pages 369–383. Springer.

Simo-Serra, E., Fidler, S., Moreno-Noguer, F., and Urtasun, R. (2014). A high performance crf model for clothes parsing. In *The Asian Conference on Computer Vision (ACCV).*

Simo-Serra, E., Fidler, S., Moreno-Noguer, F., and Urtasun, R. (2015). Neuroaesthetics in fashion: Modeling the perception of fashionability. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 869–877.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *The International Conference on Learning Representations (ICLR).*

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv*, pages 1–14.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations.*

Socher, R., Ganjoo, M., Manning, C. D., and Ng, A. (2013). Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943.

Song, Z., Wang, M., Hua, X.-s., and Yan, S. (2011). Predicting occupation via human clothing and contexts. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 1084–1091.

Sun, C., Gan, C., and Nevatia, R. (2015). Automatic concept discovery from parallel text and visual corpora. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 2596–2604.

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

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang (2015). Learning from massive noisy labeled data for image classification. *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2691–2699.

Varma, M. and Zisserman, A. (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision (IJCV)*, 62(1-2):61–81.

Vo, P. D., Ginsca, A., Borgne, H. L., and Popescu, A. (2015). On Deep Representation Learning from Noisy Web Images. *arXiv*.

Xiao, J., Ehinger, K. A., Hays, J., Torralba, A., and Oliva, A. (2014). Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision (IJCV)*.

Yamaguchi, K., Hadi Kiapour, M., and Berg, T. L. (2013). Paper doll parsing: Retrieving similar styles to parse clothing items. In *The IEEE International Conference on Computer Vision (ICCV)*.

Yamaguchi, K., Kiapour, M. H., and Berg, T. L. (2012). Parsing clothing in fashion photographs. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3570–3577.

Yang, Y. and Ramanan, D. (2011). Articulated pose estimation with flexible mixtures-of-parts. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1385–1392. IEEE.

Yoo, D., Kim, N., Park, S., Paek, A. S., and Kweon, I. S. (2016). Pixel-level domain transfer. *arXiv preprint arXiv:1603.07442*.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. (2015). Understanding neural networks through deep visualization. In *Deep Learning Workshop, International Conference on Machine Learning (ICML)*.

Yu, F. X., Cao, L., Feris, R. S., Smith, J. R., and Chang, S.-F. (2013). Designing category-level attributes for discriminative visual recognition. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. *ECCV*, 8689:818–833.

Zhang, N., Donahue, J., Girshick, R., and Darrell, T. (2014). Part-based R-CNNs for fine-grained category detection. In *The European Conference on Computer Vision (ECCV)*.

Zhou, B., Jagadeesh, V., and Piramuthu, R. (2015). Conceptlearner: Discovering visual concepts from weakly labeled image collections. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2014). Object Detectors Emerge in Deep Scene CNNs. *The International Conference on Learning Representations (ICLR)*, page 12.

Zhou, E., Fan, H., Cao, Z., Jiang, Y., and Yin, Q. (2013). Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *The IEEE International Conference on Computer Vision Workshop (ICCVW)*.