ON HANDBAG RECOGNITION AND RECOMMENDATION

YAN WANG

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2016

Acknowledgments

First and foremost, I would like to express my sincerest gratitude to my supervisor Professor Alex C. Kot, for his unfailing support and generous mentorship throughout the past four years. It is my great honor to be his student. Without him, I would never learn how to think deep and think out of box. Without him, I would never know that I need to be more careful rather than rush. I will always remember how he encourages me when I feel lost and confused. I will remember the days when we explore the research problems together. I will also remember the opportunities he provides, such as overseas exchange, FYP student supervision, master student discussion and competition participation. His charming personal characteristics, professional attitude towards research, passion for life have greatly benefit me in my future career and personal life.

I am grateful to many professionals for their helps. I especially want to acknowledge Professor Xu Dong for his effort in teaching me the machine learning skills, and introducing me to his student for discussion and collaboration. I wish to express my sincere thanks for Professor Wang Gang for offering me the opportunities to study with his group for paper reading. I would like to thank Professor Yuan Junsong, for his recommendation, advices and kindness. I would like to thank my qualifying examination panel members, Professor Anamitra Makur and Professor Jiang Xudong for the suggestions. I would like to thank Dr. Li Sheng

for his help in brainstorming ideas with me, helping to improve my scientific writing and contributing to all chapters of the thesis. I would like to thank Dr. Xu Xinxing for his patient in teaching me all the equation derivation and writing skills.

I am deeply in debt to Dr. Shen Wei for being so patient and loving during my frequent moments of stress. He has being my teacher, friend and soul mate. It is thanks to the happiness he has inspired that I was able to finish this PhD. I would also like to acknowledge Dr. Zuo Zhen and Dr. Li Yifan, for their tremendous help and support. I am thankful to the Deputy Director of ROSE lab Dr. Dennis Sng for offering me the student helper position. Thanks to all my colleagues in Workstation Resource Laboratory and ROSE lab. I would not be able to progress so fast without the help from my senior Dr. Miao Zhenwei, Dr. Wu Yuwei, Dr. Cao Hong, Dr. Yang Huijuan, Dr. Lai Jian, Dr. Chen Changsheng, Dr. Liu Siyuan, Dr. Fan Jiayuan, Dr. Chen Tao, Dr. Khosro Bahrami, Dr. Leida Li and Dr. Mu Hao. The labs would not be so harmonious without the work of Dr. Steward Chu, Mr. Joseph Lim, Ms. Wang Qian and Dr. Yang Gao. I also enjoy the time with all the group members and friends, Zhan Huijing, Lu Ze, Li Haoliang, Wan Renjie, Devadeep, Tra, Wang Xingxing, Shuai Bing, Weng Renliang, Terry, Abrar, Hu Junlin, Zhang Dajiang, Meng Jingjing, Amir, Liu Ting, Rahul, Wang Bing, Wang Li, Wang Zhenhua, Zhao Lifan, Wang Yangtao, Wang Yuan, Li Xiang, Hu Peng, Chen Jie, Yu Yi, Hu Yusong, Ye Wei, Yu Tan, and Wang Jinghua.

My last words of gratitude are reserved for my parents and sister. Thanks for their love and sacrifice since the first day I came to the world. They raise me, educate me, believe me and encourage me. This thesis would never be possible without them.

Contents

Summary

From Google to Pinterest, multimedia search engines such as Google Goggles deliver a wealth of visual information related to the search query. It recognizes and provides useful information when pointing the mobile phone camera at a business card, a book, a painting, a famous landmark, or a barcode. Vision-based techniques try to perceive and understand images by learning from the ability of human vision. Developing such techniques remains an ongoing challenge for computers.

Nowadays, multimedia systems for online advertising and commerce have a large market demand. Recent years' computer vision and multimedia communities have devoted efforts on many applications, such as fashion retrieval or recommendation for clothing, shoes, etc. Handbag has become a desirable fashion accessory, with six in ten consumers having purchased at least one new handbag in the year of 2014. Such market demand motivates the handbag recognition related vision products. However, this kind of product is still limited so far. As Google says, Goggles does not work well yet on things like food, plants, animals and some fashion items such as handbags. To develop such reliable recognition engines, we study handbag recognition and recommendation, which are key steps for building up a multimedia search system. The works in this thesis can be summarized as below.

A style-to-color discriminative representation framework for handbag recognition is carried out at first. We identify the handbag model by conducting the style-based recognition and color-based recognition sequentially due to the visual characteristics of handbags. Experiments are conducted on our newly constructed handbag datasets. The experimental results illustrate that our method achieves over 10% improvement in accuracy for recognizing handbags when compared with existing fine-grained or generic object recognition methods.

In recent years, Convolutional Neural Network (CNN) is promising for many image recognition tasks, which motivates us to design a handbag recognition algorithm based on CNN. However, after studying various CNN architectures for training the classifier, we find that the previous CNN models do not provide discriminative color information during training. Moreover, CNN models usually consider the hard label (i.e., the ground truth class label) to train a multi-class classifier. This is not sufficient especially for visually similar classes. In order to train a better CNN for classification, we present a Feature Selective joint Classification-Regression CNN (FSCR-CNN) model. It is helpful for recognizing color sensitive objects and it facilitates the classifier modeling for visually similar classes. Moreover, we propose an end-to-end handbag recognition framework. In this framework, we propose three components: (1) symmetry-based proposal localization, (2) CNN detection and FSCR-CNN classification, and (3) combination of detection scores and classification scores by conditional probability model. The experimental results verify the advantages of each component of our framework for handbag recognition.

A handbag recommendation system for e-commerce and shops is also proposed. It can help shoppers to find desirable fashion items, which facilitates online interaction and product promotion. Given the images of the shopper's preferred

handbags, the recommendation is performed by joint learning of attribute projection and one-class SVM classification. A weighted AutoEncoder method is further proposed to refine the recommended results. The results show that this scheme performs favorably based on the initial subject testing.

List of Figures

List of Tables

List of Abbreviations

WWW	World Wide Web
CNN	Convolutional neural network
FSCR-CNN	Feature selective joint classification-regression convolutional neural network
FS-CNN	Feature selective convolutional neural network
CR-CNN	Joint classification-regression convolutional neural network
CNN-G	GoogLeNet
CNN-S	Slow architecture
SVM	Support vector machine
JPO	Joint learning of attribute projection and one-class support vector machine
ReLU	Rectified linear units
ILSVRC	ImageNet Large scale visual recognition rhallenge
Deconvnet	Deconvolutional network
R-CNN	Regions with convolutional neural network features
FGVC-Aircraft	Fine-grained visual classification of aircraft
DPD	Deformable part descriptors
ROI	Region of interest
SSC	Style-specific sub-category

SPM	Spatial pyramid matching
SIFT	Scale-invariant feature transform
LBP	Local binary pattern
HOG	Histogram of gradient
SGD	Stochastic gradient descend
LOF	Local outlier factor
KDE	Kernel density estimator
UOCL	Unsupervised one-class learning
OC-SVM	One-class support vector machine
CF	Collaborative filtering
WELLSVM-SSL	Weakly labeled support vector machine - semi-supervised learning
mAP	Mean average precision
SCDR	Style-to-color discriminative representation

Chapter 1

Introduction

1.1 Background

With the development of worldwide information technology, people are easier to access useful information quickly and conveniently. Multimedia, as an important information carrier, should be paid close attention to. The evolution in this field has been changing people's traditional way of obtaining information. The function of a typical multimedia system is to get information from multiple media sources (e.g., text, graphics drawings or images) and to use it for various applications [9, 10] (e.g., multimedia search, retrieval or recommendation). An image is a visual representation of things, which is more intuitive than text. It is said that among all information transmitted to the brain, 90% is visual, and visuals are processed $60,000\times$ faster than text in the brain. An active research direction in multimedia is to build a vision-based multimedia system [11]. Given an input image, the system can understand its content and search related information on the World Wide Web (WWW). Google Goggles is a well-known image recognition mobile app that recognizes famous landmarks, paintings, books, DVDs, CDs and barcode. Given a

picture taken by a smart phone, Goggles can recognize it and return with related information.

Nowadays, such visual recognition system is one of the most exciting trends in e-commerce, especially those aiming at fashion recognition. When consumers are attracted to a product they have seen somewhere, e.g., from a fashion magazine or on the street, they usually desire to look for some information about it. However, it is often difficult to describe the product and search it in an e-commerce website. Thanks to image-based recognition, which "reads" images to identify color, shape, and texture, and can be used to identify product names or to recommend similar items. Moreover, many people carry their smart phones around which allows them to take photos of a product they desire on the fly, and search the product online. After they assess the useful properties about the desired product, e.g., model name, price, etc., they can search for more information or make a keyword-based purchase.

Fashion recognition has been studied for years. A famous fashion media company, *The Business of Fashion* once said "Anytime, anywhere, a user would be able to snap a piece of inspiration - a sharply cut coat on a passerby, or a fetching mini-dress in a magazine - and sophisticated "visual search" technology would identify and retrieve a link to the item, available to instantly buy, thereby radically shortening the path from inspiration to transaction" [12]. Fashion recognition can be employed in many multimedia applications, e.g., to be combined with other media sources to build up a practical multimedia system for online advertising and commerce. However, as *The Business of Fashion* mentioned, the current state of the recognition techniques is still not mature enough. If consumers want to search objects by image recognition, they will be disappointed.

Recommendation is also an important function in multimedia recognition sys-

tem. As people's living style is becoming conscious and time-pressed today, recommend online product that fits personal preferences would make shopping more efficient. Movies, music, news, books, research articles recommender systems have been extensively studied in recent years. Fashion recommendation is an emerging topic, while the difficulty to describe the tastes or styles of a person presents an obstacle to the progress of recommending fashion items. The recognition of fashion elements such as clothing, shoes, and makeups have been studied. Some works have been proposed to address fashion related problems, including attribute discovery from noisy web data [13], clothing parsing [14] and attribute discovery for shoes [15, 16].

In this thesis, we focus on handbag recognition and recommendation. Handbag has become a desirable fashion accessory. Six in ten consumers purchased at least one new handbag in 2014 [17]. It is said that handbag sales had risen by 10.5% to £1.16 billion in 2013. Such market demand motivates the handbag related studies, which are still limited.

Some practical applications for handbag recognition are straightforward. From consumers' view, when they are attracted to a handbag, they always ask more information regarding its specific model, price, etc. Recognizing the model of the handbag from its image can help those consumers to search for more information or even make a key-word based purchase. From manufacturers' view, getting consumers' feedback from the social network can help the handbag marketing agencies in branding purpose. While consumers usually prefer to upload photos of their precious handbags without specific model description, it will result in difficulty for marketing agencies to retrieve consumers' feedbacks based on the handbag model. By recognizing the handbags, descriptions or comments attached with the handbag photo can be easily collected. Therefore, to build up a convenient and useful

multimedia system to help users retrieve more information about the query handbag, image-based branded handbag recognition is an important step. For handbag recommendation, several scenarios can be considered. (1) A shopper clicks the handbags he/she finds interesting and the online shop provides the shopper a list of recommended handbags the shopper might prefer. (2) The handbag manufacturers recommend some new collections to shoppers according to their search history. These systems aim at finding preferred items in the collection or catching shoppers' attention when new collections are available. Therefore, it is desirable to develop such a handbag recommendation engine for e-commerce shops.

Recognizing the model of a handbag is a new instance of fine-grained object recognition that is not well addressed before. In recent years, significant efforts have been put in developing fine-grained object recognition and recommendation techniques, which are summarized as below.

1. Fine-grained object recognition: aiming at classifying visual data in a subordinate level, e.g., to differentiate blackbird from crow or to tell dandie dinmont from maltese. The main challenge of fine-grained object recognition is to differentiate fine-details among sub-categories of the same object class (e.g., birds, flowers, cars or handbags). The techniques can be roughly grouped into four categories. (1) The deformable part descriptor-based methods [18] deal with large pose variation. (2) The detection and segmentation based methods [19] decrease the impact of background. (3) Human-interactive methods [20] incorporate human intelligence to assist the recognition. (4) Deep Convolutional Neural Network (CNN) based methods [21] transfer the CNN models trained on large labeled dataset (e.g., ImageNet [22]) to specific visual recognition tasks.

2. Recommender systems: aiming at predicting the rating or preference that a

user will prefer from a collection of things. Much research for recommender system is centered on different domains including movies [23], music [24], article [25], etc. Typically, recommender systems provide recommendations via collaborative filtering or content-based recommendation [25]. Some e-commerce websites consider collaborative recommendation, and it works well when enough ratings are given by a large community of shoppers [26]. Content-based recommendation recommends an item to a user based on the item description and a profile describing the user's interests. Some image-based fashion recommendation systems are also proposed, such as shoe recommendation [27], clothing recommendation [28] and clothing-accessories recommendation [29].

These techniques achieve promising results in different aspects, however, there are still many issues needed to be addressed when it comes to handbags. Fine-grained object recognition methods are not directly applicable to handbags. Specifically speaking, handbags do not have prototypical regions or annotated parts, which brings difficulties when seeking for a solution by using deformable part descriptor-based methods. Segmentation based recognition methods would not perform well if the foreground region were not segmented correctly. Human-interactive methods require human labor, which is a burden for users. Current CNN methods do not provide discriminative color information during training. Moreover, CNN models usually consider the hard label (i.e., the ground truth class label) to train a multi-class classifier. They face some problems when differentiating visually similar classes. As for recommendation systems, collaborative filtering generally outperforms content-based recommendation. However, when it encounters the problems like early rater (items that are first appear without user ratings), sparsity (the user-item matrix that is sparse) or gray sheep (individu-

als who do not consistently agree or disagree with others in a small community of users), the collaborative filtering does not perform well [24]. Current content-based techniques sometimes recommend items which share the most concrete attributes (e.g., color, pattern, weather, etc.) with the shopper's data. However, they somewhat ignore important problems on how to define the attributes or how to match with shoppers' desire.

1.2 Objectives and Major Contributions

The research carried out in this thesis focuses on handbag recognition and recommendation. The objective lies in two aspects: (1) recognizing the handbag model, given a branded handbag image as the input and (2) recommending handbags, given shopper-clicked handbag images as the input.

The major contributions of this thesis can be summarized as follows:

1. **Discriminative representations for handbag style and color**: A new branded handbag recognition problem is studied. In order to deal with inter-class style (including shape, pattern and texture) similarity and intra-class color variation, patch-based discriminative representations for handbag style and color are proposed. In order to evaluate our algorithms, we create a new branded handbag dataset, named as BrandBag, covering around 10,000 images from 401 models of two handbag brands[1][2] (named as BrandBag-I and BrandBag-II, respectively). Compared with BrandBag-I dataset, handbag images belonging to the same model in BrandBag-II dataset have more variations in illumination, deformation or viewpoints. Patterns such as checkerboard and floral pattern

[1] `http://www.louisvuitton.eu/front/\#/engE1/Collections/Women/Handbags`
[2] `http://www.coach.com/shop/women-handbags?viewAll=true`

with brand initial are universal in BrandBag-I, while most handbag models in BrandBag-II dataset have plain pattern. Experimental results show that compared with those public available patch selection techniques [4, 30], the proposed patch selection is more suitable to extract discriminative information for recognizing handbags. It also shows that the proposed framework works favorably for handbag recognition.

2. **Feature selective joint classification-regression learning for handbags**: A CNN-based branded handbag recognition framework is designed. In the deep CNN level, the proposed Feature Selective joint Classification-Regression CNN (FSCR-CNN) has two innovations. (1) Feature selective CNN (FS-CNN) incorporates discriminative color information to classify color sensitive objects. (2) Joint Classification Regression-CNN (CR-CNN) assigns a distribution to measure how similar this class to other classes to improve the discriminability of the learned model. In the system level, an end-to-end framework which recognizes the model for a given input handbag image is designed. Experiments on the newly constructed handbag dataset verify that the proposed FSCR-CNN can boost the performance by at least 8% in accuracy. It can also improve the recognition accuracy by over 5% for other fine-grained objects on different CNN architectures. An additional comparison of style-to-color discriminative representation (short for SCDR) and FSCR-CNN for handbag recognition is conducted. The FSCR-CNN classification model performs better than the SCDR framework on the BrandBag dataset, with 3% improvement in accuracy. However, FSCR-CNN has higher requirements on resources (i.e., GPU).

3. **Joint learning of attribute projection and one-class SVM classification (JPO) for handbag recommendation**: A handbag recommendation

framework is proposed to recommend the desirable handbags for a shopper, given shopper-clicked handbags. We learn a projection that can automatically project the original features onto an attribute space, and meanwhile learn a hyperball to separate the learned attribute features from the origin. Experimental results on the initial subject testing show that the distances among newly learned features (potential attributes) are closer and the new classifier is more reliable to differentiate the positive data with others by the joint learning.

1.3 Organization of the Thesis

This thesis is organized as follows:

In Chapter 2, the feature and classifier learning techniques are first introduced, a brief review on current fine-grained object recognition and recommender systems is presented.

In Chapter 3, a novel discriminative representation for handbag style and color technique is proposed to recognize the model of a handbag.

In Chapter 4, a novel feature selective joint classification-regression CNN model is proposed for training the classifier. An end-to-end framework in the system level is further proposed to recognize the model for a given input handbag image.

In Chapter 5, a handbag recommendation system is proposed for recommending desirable handbags for a shopper, given the shopper's preferred handbags. A post-processing of anomaly detection is also proposed for further improving the recommendation performance.

In Chapter 6, the conclusions are drawn and some future research directions are suggested.

Chapter 2

Literature Review

In this chapter, we review the related works, including hand-crafted features extraction, traditional classifier learning, deep learning, fine-grained object recognition and recommendation system. We also discuss the difference between the methods proposed in this thesis and the related works.

2.1 Feature and Classifier Learning

Machine learning is a technique to enable computers to learn without being explicitly programmed. The typical workflow of machine learning is illustrated in Fig. 2.1. It consists of two phases: training and testing. During training, two kinds of data can be input into the system: supervised and unsupervised. Then through certain feature extraction procedure, their features are fed into machine learning algorithm to either group these data or output a predictive model. In the testing phase, when new data come, the machine learning system will output their annotations. In this thesis, we focus on handbag recognition and recommendation, which is fitted into the workflow. In the next part of this section, we will discuss about the feature and classifier learning techniques in recent years.

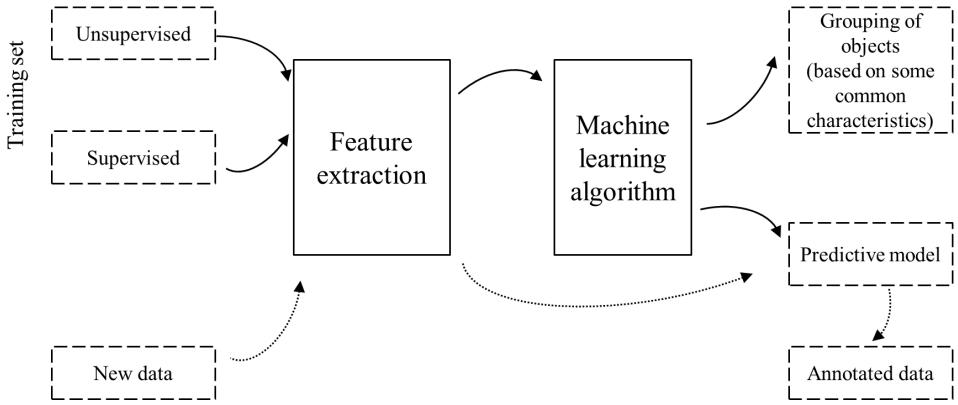


Figure 2.1: Typical Machine Learning workflow.

2.1.1 Hand-Crafted Feature Extraction and Classifier Learning

In the last decade, numerous hand-crafted features have been developed to extract robust, informative and compact image representations [31–33]. Various features have been proposed for recognition tasks, including SIFT [31], HOG [34] and LBP [32]. SIFT measures the appearance of a point by computing the weighted spatial histogram of gradients in its neighborhood. It is robust to rotation and illumination. HOG captures the intensity gradient structure and edge direction of the local shape with uniform sampling and fine orientation binning. Orientation based features like SIFT and HOG are robust to changes in brightness. LBP is used to analyze the texture which labels the pixel through utilizing the gray scale contrast of the neighborhood and this pixel. It thresholds the neighboring pixel and saves the result as a binary number [35]. It is shown to perform well under some monotonic gray-scale changes. Some other features like rotationally invariant Maximum Response filter sets (Texton) are also shown to perform well for discriminating isotropic as well as anisotropic textures [36].

To obtain a compact image representation, local image descriptors are clustered

into a visual vocabulary. Based on the visual vocabulary, any local image descriptor can be quantized into a feature vector. Eventually, the feature vectors of a local region are pooled together for classification. This is the well known bag-of-features image representation method [8, 37]. This method is simple, computationally efficient and intrinsically invariant. The improved fisher kernel proposed by Perronnin *et al.* [38] deals with the lossy process when conducting quantization. It encodes additional information about the distribution of the descriptors when doing clustering. Spatial Pyramid Matching (SPM) proposed by Lazebnik *et al.* [8] is an extension of bag-of-features. It studies the spatial layout of features and represent images at several levels of resolution based on pyramid match kernels.

Classifiers such as Support Vector Machines (SVM) and random forest are well-known and widely used. The goal of SVM [39] is to find the decision boundary which separates the data of two classes with maximum margin. Typical random forest [40] builds many classification trees. A learned class distribution is associated with each leaf node of the tree. An input feature vector falls into a leaf node of a decision tree by recursively branching left or right down, according to the learned function.

2.1.2 Deep Learning

Deep learning is a new area of machine learning research. It models high-level abstractions or concepts in data by using hierarchical networks [41]. Deep learning includes many variants. The deep neural networks are artificial neural networks with multiple hidden layers between the input and output layers. The deep belief networks build multi-layer generative models of unlabeled data by learning one layer of features at each time. In this section, we review two deep neural networks:

the convolutional neural network and stacked auto-encoder.

2.1.2.1 Convolutional Neural Network (CNN)

Recently, CNN architecture shows promising performances in a lot of computer vision applications such as image recognition, image understanding, etc. [1, 42, 43]. Several CNN structures are proposed to learn discriminative features from raw image inputs and exhibit hierarchical semantic information along their deep architecture [1, 42–46].

As shown in Fig. 2.2, the popular "AlexNet" ImageNet model [1] builds an eight-layer architecture. The first five layers are convolutional layers and followed by three fully connected layers, with a softmax as the output. More specifically, the input of the CNN is a 3-channel image, which is first filtered by five convolutional layers. Followed by the 1^{st} and 2^{nd} convolutional layers are the response-normalization layers. Max-pooling follows the 5^{th} convolutional layer and the response-normalization layers. Rectified linear units (ReLU) is adopted after each convolutional and fully-connected layer. Convolutional layers provide various feature maps, while the max-pooling layers reduce computation for consecutive layers and provides a form of translation invariance. ReLU increases the nonlinear properties of the decision function. The performance of "AlextNet" is more than 10% better when compared with traditional methods in the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC-2012) competition.

After "AlexNet", an integrated framework OverFeat [44] was proposed to train a convolutional network which can simultaneously do the task of object detection, localization and classification in images. It wins the localization task of ILSVRC-2013. OverFeat explains how convolutional networks used for localization and detection. It says that convolutional networks are inherently efficient when ap-

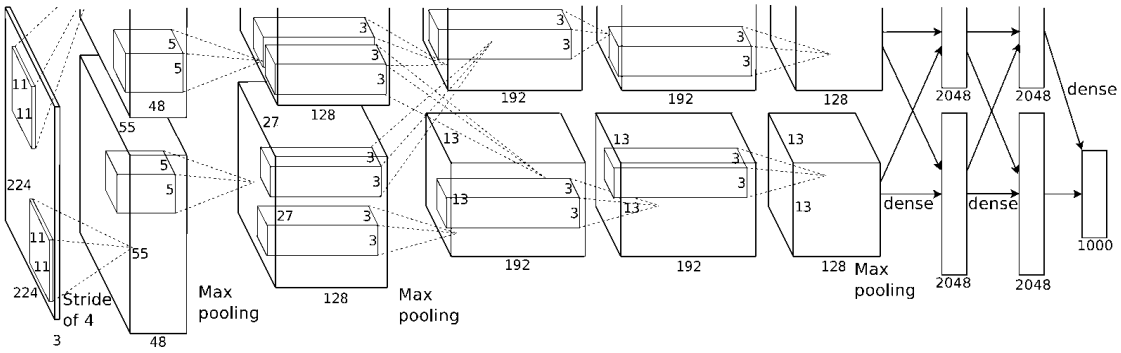


Figure 2.2: "AlexNet" architecture, extracted from [1].

plied in sliding windows because they share computations common to overlapping regions, which can be updated efficiently by gradient descend.

Zeiler and Fergus [42] proposed a Deconvolutional network (deconvnet) to visualize the intermediate feature layers and give insight into the operation of the classifier. The lower layers show edges, textures, while the higher layers show greater invariance and exaggeration of discriminative parts of the image.

Chatfield *et al.* proposed CNN-S architecture [47], which is related to OverFeat. Compared with OverFeat, less filters are applied in the 5^{th} convolutional layer and a local response normalization layer is added after the 1^{st} convolutional layer rather than contrast normalization. It achieves better performances on many image classification tasks.

To address the object detection problem, Girshick *et al.* [48] proposed regions with CNN features (R-CNN). It localizes objects by applying CNN to bottom-up region proposals. "Recognition using regions" [49] paradigm is adopted in this work. With image-level annotations (i.e., no bounding box labels), the CNN is trained on the ILSVRC 2012 dataset. For any query during testing, it first generates around 2000 region proposals [3]. For each of the proposals, a feature vector is extracted using the trained CNN. Then category-specific linear SVMs are

further adopted to classify each region.

Previous studies show that the most straightforward way of improving the performance of deep neural network is by increasing their size. This simple solution has two drawbacks. (1) A larger number of parameters make the enlarged network more prone to over-fitting, especially if the number of labeled examples in the training set is limited. (2) Increased network size will result in increased use of computational resources. To deal with these issues, a new 22 layers deep network GoogLeNet [43] is proposed recently, which increases the depth and width of the network with the computational budget unchanged. It achieves the best results for classification and detection in the ILSVRC14.

2.1.2.2 Stacked AutoEncoder

AutoEncoder is a feed-forward neural network trained to reproduce its input at the output layer, as shown in Fig. 2.3. It has several hidden layers. These hidden layers are composed of multiple encoder layers and decoder layers. "Encoder" network transforms the high-dimensional data into a low-dimensional code and "decoder" network recovers the data from it. It has many applications. For example, the denoising AutoEncoder [50] feeds a "noisy" data into an AutoEncoder, and to produce the denoised version after training. Another application is learning deep invariant features [51], which learns the outputs corresponding to the inputs that are neighbors to be nearby, and vice versa. Adopting Stacked AutoEncoder on face landmark estimation is also popular. The successive Stacked AutoEncoder [52] characterizes the nonlinear mappings from face image to face shape in different scales based on the shape predicted from previous AutoEncoder.

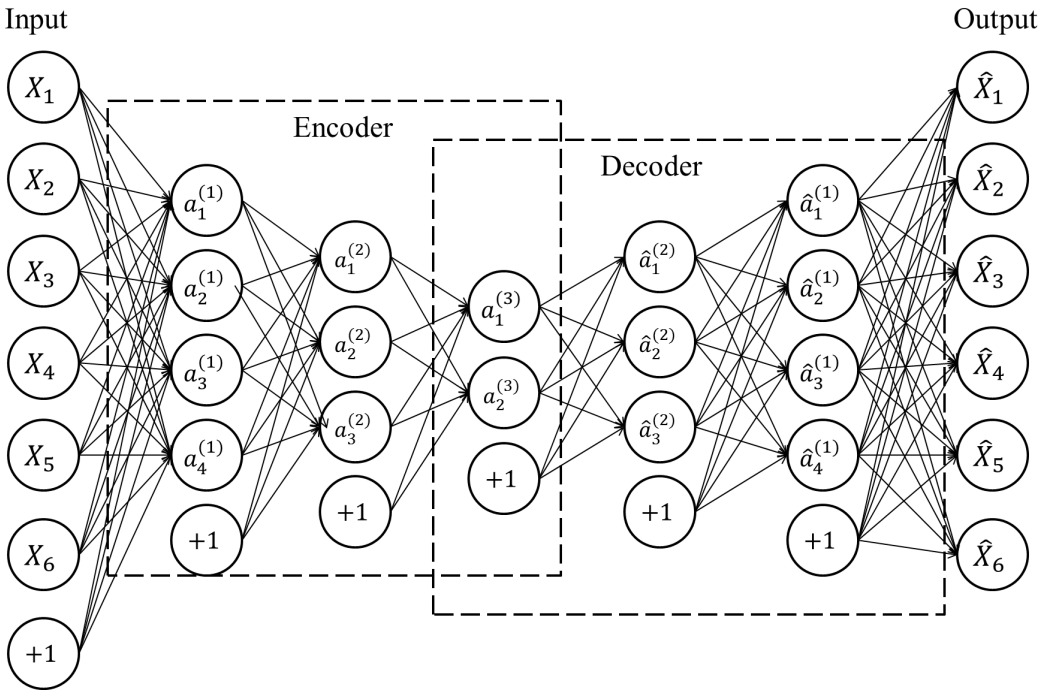


Figure 2.3: Stacked AutoEncoder [2].

2.2 Fine-Grained Datasets & Classification Strategies

Fine-grained object recognition is different from generic object recognition. It refers to the task of recognizing the subcategory under the same basic category such as flower species or handbag models. Studies show that even human beings cannot recognize some subcategories [7], which makes the problem challenging and valuable.

2.2.1 Fine-Grained Object Datasets

Several competitive benchmarks have been built for the research of fine-grained object recognition. One of the most challenging datasets is Caltech-UCSD Birds Dataset [7, 18, 21, 53]. It contains 11,788 images of 200 bird species. A total of

(a) (b)

Figure 2.4: Examples of (a) California Gull and (b) Glaucous-winged Gull.

5994 images are used for training and the rest 5794 is for evaluation. Bird images in this dataset is collected from Flickr. Images are annotated with bounding boxes, part locations and attribute labels such as breast pattern and tail shape. Birds in the dataset have arbitrary poses.

Another challenging dataset Stanford Dogs Dataset [6] contains over 20,000 annotated images of 120 breeds of dogs. The images and annotations are collected from ImageNet [22]. Dogs inside one class could have different ages, poses, or colors. Some dogs even wear clothes or heavily occluded.

Oxford Flowers Dataset [5] was built in early ages. This dataset consists of 102 different flower categories common to the UK, covering 40 to 258 images per category. Those images are with different scales, resolutions, lighting conditions and clutters. Most of them are downloaded from the web, while some images are taken by the authors.

Other datasets are also popular. Oxford-IIIT-Pet Dataset [54] contains 7349 images of cats and dogs over 37 different breeds. Pittsburgh Fast-Food Image Dataset [55] was created for dietary assessment, with 101 types of food harvested with 4545 images. Cars Dataset [56] contains 16,185 images of 196 classes of cars,

with classes at the level of Make, Model, Year. Fine-Grained Visual Classification of Aircraft (FGVC-Aircraft) [57] contains 10,200 images of aircraft. It is used as part of the fine-grained recognition challenge FGComp 2013. These benchmarks offer great convenience for researchers who are doing fine-grained object recognition problems.

2.2.2 Fine-Grained Object Recognition Methods

In this section, a brief literature review on current fine-grained object recognition techniques is given. The pros and cons of different techniques are discussed.

Fine-grained object recognition demands the algorithm to differentiate the subtle difference among these subcategories, which is an emerging topic in recent years [58–61]. There are mainly two challenges for fine-grained object recognition. (1) Small inter-class variation. For example, the California Gull and Glaucous-winged Gull are very similar in appearance, but differ in the color of the beak and shape of the legs, as shown in Fig. 2.4. (2) Large intra-class variation due to the changes of illumination, pose, occlusion, and even color. Various techniques are proposed to tackle with the challenges.

2.2.2.1 Discriminative Part/Pose Localization and Representation

Discriminative parts localization methods are very popular. These methods are motivated by the observations that some semantic parts have isolate subtle appearance differences among fine-grained subcategories. Thus, keeping the discrimination facilitates fine-grained categorization.

Farrell *et al.* [62] presented a volumetric poselet scheme, and used seven-parameter ellipses to represent pose-normalized appearance for bird classification. In the work of Parkhi *et al.* [54], deformable part model [63] is adopted to de-

tect pet's face for recognizing cats and dogs. Liu *et al.* [64] proposed to build a breed-specific exemplar model to represent dog breeds and their face parts for dog recognition. An alignement-based fine-grained categorization method was proposed by Gavves *et al.* [65, 66]. In their proposed supervised alignments, the ground truth locations of basic object parts, such as the "beak" or the "tail" of birds, are available for training. The alignments are then transferring part annotations from training images to test images. In 2013, Zhang *et al.* [67] proposed two deformable part descriptors (DPD) based on the deformable part model. One is DPD-strong, which pools semantic part features across different components. The other is DPD-weak, which solves the shortcomings of the strong classifiers and computes pose-normalized descriptors. Then the following year, the same group of authors proposed to learn whole-object and part detectors for the recognition, such that no object bounding box is required during testing [68]. A pose normalized deep convolutional net for recognizing bird species was proposed by Branson *et al.* [18]. In this work, a set of prototypical regions are learned based on the keypoint annotations. Then deep convolutional nets are trained on different alignment models by fine-tuning. Eventually, features extracted from multiple layers are concatenated and input into an SVM to do the classification.

The advantage of semantic part-based representation for fine-grained object recognition is explicitly isolating subtle appearance differences associated with specific object parts. For the objects which do not have prototypical regions, annotated parts or keypoint annotations, the semantic part-based methods may struggle.

Rather than learning semantically meaningful parts, some part-based representations learn discriminative parts in an unsupervised manner. A popular and publicly available fine-grained object recognition system is proposed by Yao *et al.*

[4]. In this work, a discriminative and randomized patch selection algorithm, random forest [40] with strong classifiers, is designed to find image regions that contain discriminative information through a dense feature space. Duan *et al.* [69] discovered localized machine-detectable and human-understandable attributes from object bounding boxes. This method learns semantically meaningful regions and mines text from the learned distinct regions such as "red eye" or "blue wings" for bird species. A strategy which integrates feature learning and unsupervised part discovery was first proposed by Krause *et al.* [70]. This strategy adopts the CNN to train features and discover sets of aligned images with similar poses. The fully unsupervised part discovery technique performs well for objects such as cars. Since cars contain distinct parts that can give information about the overall shape or type of the car, e.g., whether the car is a Sedan or SUV. Zhang *et al.* [71] proposed a superpixel pyramid for each image, and developed a dense graph mining called graphlets to localize multiscale discriminative object parts for each category. For those techniques, without the semantic meaningful parts information, how to find the discriminative parts effectively is an important issue.

2.2.2.2 Segmentation/Detection with Classification

Segmentation/detection with classification methods show that segmenting out the background distracters is beneficial. It helps classification in several ways, such as localizing the objects. Nilsback and Zisserman [5] proposed a segmentation scheme. It learns non-class specific foreground and background color distribution by manually labeling pixels (i.e., part of the flower or part of the greenery) in the training images. Based on the segmented flower regions, three types of features are extracted and combined for classification. Chai *et al.* [19] proposed a fine-grained visual categorization method. It uses both segmentation and part

localizations to encode the image content into a highly-discriminative visual signature. In the work of Angelova *et al.* [19], the low-level regions likely belonging to the object is first detected. Then a propagation based full-object segmentation is performed. The recognition performance is significantly improved by combining the segmented object with the classification algorithm. Wang *et al.* [72] incorporated the information from saliency aware object detection. Based on the location information, an object-centric sampling scheme was proposed first. Then a CNN is trained for classification. Angelova and Long [73] designed an automatic segmentation model based on superpixels and Laplacian propagation. Then a region pooling technique doing the fine-grained categorization was proposed. For those segmentation/detection with classification methods, usually the classification results highly rely on the segmentation, while the segmentation accuracy cannot be guaranteed.

2.2.2.3 Human-Interactive Assistance

Human-interactive methods incorporate human intelligence to assist the recognition. Branson *et al.* [74] designed a system, which asks users visual questions such as "Is the belly black", along with computer vision algorithms. This method is a feasible way to recognize bird species, especially for non-experts. Deng *et al.* [20] proposed an online "Bubbles" game which includes human-in-the-loop to select discriminative features. At each round of the "Bubbles", the computer shows two bird species to the player, where the images are burred so that the player can only observe rough outlines. In this game, a so-called successful human player clicks the circular regions which contain discriminative features. Based on the regions, a "BubbleBank" representation formed by the computer can be further used for learning classifiers. Rejeb Sfar *et al.* [75] presented a semi-automated system con-

taining two modes of operation for fine-grained categorization and plant species identification. (1) Given a test image as input, instead of providing a single estimate, the system outputs a set of candidate species. So that the user needs to do the final identification. (2) The user is required to mark the key-points of the leaf at the beginning to facilitate the recognition. He also needs to identify the species through a set of candidates provided by the system in the end.

These human-in-the-loop methods bring a burden for users, especially those unprofessional ones. Therefore, these methods always try to achieve a balance between the accuracy and the human burden.

Recently, some CNN-based techniques are also proposed. Xiao *et al.* [53] combined object-level attention and part-level attention to train domain-specific deep nets. This work tries to detect objects and their parts in the annotation-free scenario. Azizpour *et al.* [21] investigated the transferability of ConvNet representation for a particular target task from aspects such as network width, network depth, and dimension. It optimizes the transferability factors and improves the visual recognition results. These techniques do not consider the network configuration of color selection and label distribution learning when it comes to recognizing handbags.

2.3 Recommender System & Fashion Recommendation

Recommender systems have become popular these years. Applications like movie recommendation, music recommendation, book recommendation or article recommendation are studied extensively in data mining or information retrieval communities. Image-based fashion recommendation is on the rise due to its practical

application in e-commerce. In this section, we give a brief literature review on the recommender systems.

2.3.1 Recommender System

In order to develop an effective recommender system, user preferences need to be learned. Different types of user preferences can be captured via

1. Explicit feedback: user ratings about the products. It offers explicit positive and negative feedbacks. However, in the real scenario, the user-item matrix is always sparse. We can hardly obtain reliable information from user-item pairs in the sparse utility matrix.

2. Implicit feedback: whether a user viewed a particular content. It indirectly reflects a user's true opinion of the content item. Therefore, it is not easy to identify negative signals. However, this implicit feedback can be generalized more easily.

2.3.1.1 Recommender System Datasets

Several practical and public available datasets have been built for recommendation. For movie recommendation, most datasets are with explicit feedbacks. Datasets like MovieLens 100k, MovieLens 1M, MovieLens 10M, MovieLens Latest and MovieLens Tag Genome are popular [76]. MovieLens Last is the largest dataset among them, which is with 21,000,000 ratings and 510,000 tag applications applied to 30,000 movies by 230,000 users. Tag Genome contains 11 million computed tag-movie relevance scores from 1,100 tags applied to 10,000 movies. For music recommendation, datasets with implicit feedbacks are collected. Last.fm [77] has implicit feedbacks from 360,000 users for 500,000 songs. Echo Nest Taste

Profile Subset [77] used in the Million Song Datastet Challenge 2012, has implicit feedbacks from 1 million users for 380,000 songs. For book recommendation, one book-crossing dataset [78] with 278,858 users providing 1,149,780 ratings (explicit/implicit) about 271,379 books was collected.

2.3.1.2 Recommender System Technologies

The technologies used in the recommender system can be categorized into two broad groups: content-based method and collaborative filtering. In this section, we give a brief literature review on these technologies and discuss the advantages and limitations of them.

Content-Based Method The key idea of the content-based approach is to recommend items upon a profile of the user's interests and the item description. Each item is represented by some characteristics of the item that are discovered. For example, the features of a movie are relevant to actors, director, the year, the genre or general type of the movie. Several works are proposed to represent items. For example, keyword-based vector space model presented by Li *et al.* [79] extracts multi-dimensional keywords and calculate the corresponding preference/weight. Wang and Blei [80] proposed an ontology based semantic analysis method. They designed a topic model to present articles in terms of latent themes. Saxina *et al.* [27] proposed an image-based shoe recommendation system. It assesses a dataset of 13,000 men's shoes and extracts the shape, color and texture of them. Then based on these features, the system is able to recommend shoes similar to a user's input.

Another line of research is to learn user profiles. This will learn a relevance score of each item and rank items according to the user preferences. Tu and Dong

[81] proposed a personalized fashion recommendation system, to help customers find their suitable fashion items. Three models were proposed in the system. (1) Interaction and recommender model associates consumers' personalized preference. (2) Evolutionary hierarchical fashion multimedia mining model extracts the main component of the fashion information. (3) Color tone analysis model relates the main color tone of the skin and the clothes. Liu *et al.* [28] proposed a clothing recommendation system, and suggested suitable clothing by attributes for a given user-input occasion. Concretely, they proposed a latent SVM based recommendation model. This model uses middle-level clothing attributes such as clothing category, color and pattern as latent variables. Ajmani *et al.* [82] proposed a new ontology-based recommendation, which encodes subjective knowledge of experts. Such that the algorithm can tell whether the visual properties of the chosen dress blend well with the occasion. In this work, Ajmani *et al.* represented the visual personality of an individual through some concepts like the color season and the body shape. More specifically, the Color Season model [83] and a probabilistic causal model are used to relate suitable garment colors with the color season of a person. Huang *et al.* [84] proposed a user-specific clothing image recommendation system. They designed an active learning based clothing image retrieval technique, and incorporated user-interaction during training. One recent clothing and accessories recommendation system proposed by McAuley *et al.* [29] recommends items that match with other items, e.g., recommend a matching shirt for a pair of jeans. This system models human visual preferences, and trains the recommendation model on pairs of objects offered by Amazon.

Collaborative Filtering Collaborative filtering is to recommend items that people with similar tastes or preferences liked in the past [85]. When a large num-

ber of users and their feedback are available, collaborative filtering can determine similar users (e.g., users who buy both pizza and salad for meal), and then recommend items among them (e.g., recommend one user coke since the other user who buys the same pizza and salad buys coke). Meanwhile, collaborative filtering can also determine the similarity between items, i.e., whether the items usually purchased by the same user.

There are basically two categories of techniques for collaborative filtering: neighborhood method and matrix factorization. Neighborhood method evaluates the preference of a user for an item, by using the ratings of other users who have similar rating patterns for this item. Baltrunas and Ricci [86] mentioned that contextual information was not exploited by collaborative filtering if the user's rating can be influenced by different contextual conditions. Therefore, they proposed to split an item into two items, such as to split "a place to visit" into "this place in summer" and "this place in winter". Adamopoulos [87] proposed a probabilistic neighborhood-based method. This method includes a new probabilistic method for neighborhood selection which considers the similarity levels between the target user and similar users. Verstrepen and Goethals [88] used only implicit feedbacks, unifying user- and item-based nearest neighbors algorithms in collaborative filtering. This method incorporates the most well known user- and item-based algorithms.

Matrix factorization maps the user feature and the item feature into a joint latent factor space. It models user-item interactions as a dot product between those feature vectors. Luo *et al.* [89] developed a non-negative matrix-factorization based collaborative filtering model. Existing matrix factorization methods assume that user preference is consistent among all the products that this user has rated. Kabbur and Karypis [90] modeled the users with multiple interests, and combine

both global preferences and interest-specific preference of the users to obtain better performance.

Content-based method and collaborative filtering have advantages and drawbacks respectively. For content-based recommendation method, its advantages are:

1. It exploits only preferences provided by the user to build his/her own profile, which is user independent. No user-item matrix is required.
2. It is transparent, since explanations can be given by listing content features or descriptions about an item and illustrates why the system works.
3. It is able to recommend an item that is not rated by any user, such that when any new items come, the system can still recommend them.

Its limitations lie in:

1. Domain knowledge is always required to obtain appropriate content analysis.
2. For a new user, with no ratings or preference record, reliable recommendations will not be provided by the system.

As for collaborative filtering method, its advantages are:

1. No content information is required or elaborately designed.
2. Serendipitous or unexpected items can be recommended by observing similar-minded people's behavior.

While its limitations are:

1. It cannot produce recommendations if there are no ratings available. For any item when it first appears, it is less likely to be suggested to users.
2. In the social network, the user-item matrix is usually very sparse, even around 99% sparsity.
3. For a small or medium community of users, the opinions of some individuals do not always agree or disagree with other groups of people. Therefore, collaborative filtering would not provide benefit for these individuals.

2.4 Summary and Discussions

In this chapter, the basic feature and classifier learning techniques are introduced. A brief review on various fine-grained object recognition and recommendation is conducted afterwards. It can be seen that the researchers have devoted a lot of efforts in developing fine-grained object recognition techniques and frameworks in recent years. Some are domain-specific solutions, while some are general. Handcrafted features and classifiers are adopted for some work as they can be explicitly designed to capture domain knowledge, such as the poselet methods proposed for animals. Handbag is a new instance of fine-grained object recognition. According to our observation, the styles of some handbag models are very similar, their discriminability is shown from (1) small decorations on local parts and (2) subtle texture difference. Another observation is that, generally the handbags of a brand are designed with diverse styles, and those within the same style only differ in color. These observations motivate us to propose a novel handbag framework in Chapter 3, which extract discriminative handbag representations, for style and color respectively. Two representations are proposed for style recognition. (1) Discriminative patch selection serves as a supervised mid-level visual representation. It is motivated by Yao's work [4], reviewed in Section 2.2.2.1. However, compared to Yao's work, the difference is obvious. Instead of sampling patches densely and searching discriminative ones through the image, we compare patches from different images with the same location. It can capture the handbag discriminative information in an efficient manner. (2) Complementary feature extraction serves as an unsupervised low level visual representation for handbag style. For color representation, the most dominant color feature elements are selected for recognition handbag models. The proposed framework requires small memory and can

be trained fast on CPU.

CNN is promising for many computer vision tasks. It motivates us to adopt CNN for handbag recognition. Previous CNN models do not explicitly incorporate discriminative color information and only adopts the ground truth class label to train the classifier. Based on this study, we propose a feature selective joint classification-regression CNN (FSCR-CN) in Chapter 4. FSCR-CNN incorporates discriminative color information and assigns an additional soft label during training. It outperforms CNN for classifying color sensitive objects and visually similar classes.

Previous fashion recommendation systems mainly focus on clothes or shoes. None of the research focuses on image-based handbag recommendation. In Chapter 5, we proceed the fashion recommendation by assessing handbags. Due to the sparsity of user-item matrix and fast pace of new handbag designs appearing in the market, we do not adopt collaborative filtering for our problem. We assume that consumers always prefer handbags with similar attributes during a certain period of time, while the attribute is difficult to define. Based on this, we propose a Joint learning of attribute Projection and One-class SVM classification (JPO), given shopper-clicked handbags. Our focus in this work is to capture more common properties of handbags clicked by each shopper. In the same chapter, we also propose a post-processing: weighted AutoEncoder, to deal with the large fraction of anomaly in the recommended handbags. It further refines the handbag recommendation results.

Chapter 3

Style-to-Color Discriminative Representation for Handbag Recognition

In this chapter, we deal with handbag recognition [1]. In literature, there are a few techniques proposed for fine-grained object recognition. However, those techniques cannot achieve satisfactory performance on the handbag recognition problem. Handbag recognition is challenging due to the inter-class style (including shape, pattern and texture) similarity as well as the intra-class color variation. We focus on the discriminative representations for handbag style and color. For handbag style representation, two supervised mid-level patch selection procedures are proposed to select discriminative patches, regarding individual classes and pairwise classes. We also propose a low-level complementary feature, extracted from texture enhanced mid-level patches, to capture the fine details of the mid-level patches. For handbag color representation, we propose to extract dominant color features to handle the illumination changes. The performance of our proposed method is evaluated on a newly built branded handbag dataset. The results show that our method performs favorably in recognizing handbags, with over 10% im-

[1] Part of this chapter was presented in [91, 92]

Figure 3.1: Illustrations of the main challenges in handbag recognition. (1) Inter-class style similarity: In each black box of (a) and (b), we show two handbags of different models, which only have subtle differences in style. (2) Intra-class color variation: In each row of (c), we show four handbags of the same models. The styles of all three handbag models are the same, while their appearances differ in color.

provement in accuracy when compared with the existing fine grained or generic object recognition methods.

3.1 Introduction

The explosive usage of personal devices with high resolution cameras have made visual object search more and more popular in our daily life, especially on the mobile end. Various researches have been done on visual object search, including user-centric mobile display [93], mobile video surveillance [94], multimedia retrieval [95–100], fashion search [101], fashion recommendation [28, 102], fashion parsing

[14], and landmark recognition [103].

Image-based branded handbag recognition is an interesting while challenging problem, which caters people's pursuit for fashion. As introduced in Chapter 2, a growing literature corpus has proposed various techniques for fine-grained object recognition. A lot of methods deal with domain-specific problems [5, 54, 74, 104]. In these works, the information predicted on a specific object category is explored. For example, deformable part model or pose normalization techniques are commonly employed for bird, cats and dogs recognition [18, 54, 59], because they have semantic keypoint locations such as head or body. Segmentation algorithms are promising for plants like flower or leaf recognition [5, 104–107]. These methods motivate us to study the visual characteristics of branded handbags.

For simplicity, the "branded handbag" will be termed as "handbag" for short in the rest of the chapter. The challenges of handbag recognition lie in the following two parts:

1. Inter-class style (style is referred to shape, pattern and texture) similarity, where two major problems are needed to be solved for this challenge. Firstly, the styles of some handbag models are very similar, only small decorations on local parts show discriminability, as shown in Fig. 3.1 (a). Secondly, subtle texture difference between handbag models is non-trivial for the recognition. However, the texture is always not distinct due to the lighting condition or out of focus, such as the visually similar handbags but with different embossed texture patterns shown in Fig. 3.1 (b).

2. Intra-class color variation: generally the handbags of a brand are designed with diverse styles, and those within the same style only differ in color. However, the illumination changes enlarge the intra-class color variance. Fig. 3.1 (c) gives

some examples for handbag models sharing the same style.

In this chapter, we aim to address the aforementioned challenges. Due to the visual characteristics of handbags, we identify the handbag model by style and color sequentially. To recognize the handbag style, firstly, two supervised mid-level discriminative handbag patch discovery strategies are proposed to separate visually similar handbag styles. Secondly, we propose to extract the low-level feature SIFT [31], HOG [34] or LBP [32] from a texture enhanced handbag region, so as to complement the details that are not captured by the original feature extracted directly from the original handbag region. The handbag models sharing the same style, named a Style-specific Sub-Category (SSC), are then identified by the dominant color feature elements, selected through a supervised manner.

As there is no benchmark available for handbag recognition, we build a branded handbag dataset for evaluation. In terms of handbag recognition, our proposed method is shown to be able to achieve over 10% improvement in accuracy when compared with the existing fine grained or generic object recognition methods [4, 33, 108].

We organize the rest of the chapter as follows. Sec. 3.2 illustrates our designed technique for handbag recognition. Sec. 3.3 introduces the dataset construction procedure and Sec. 3.4 presents the experimental results as well as some discussions, followed by the summary in the last section.

3.2 Style-to-Color Discriminative Representation for Handbag Recognition

Fig. 3.2 shows the pipeline of our proposed handbag recognition framework.

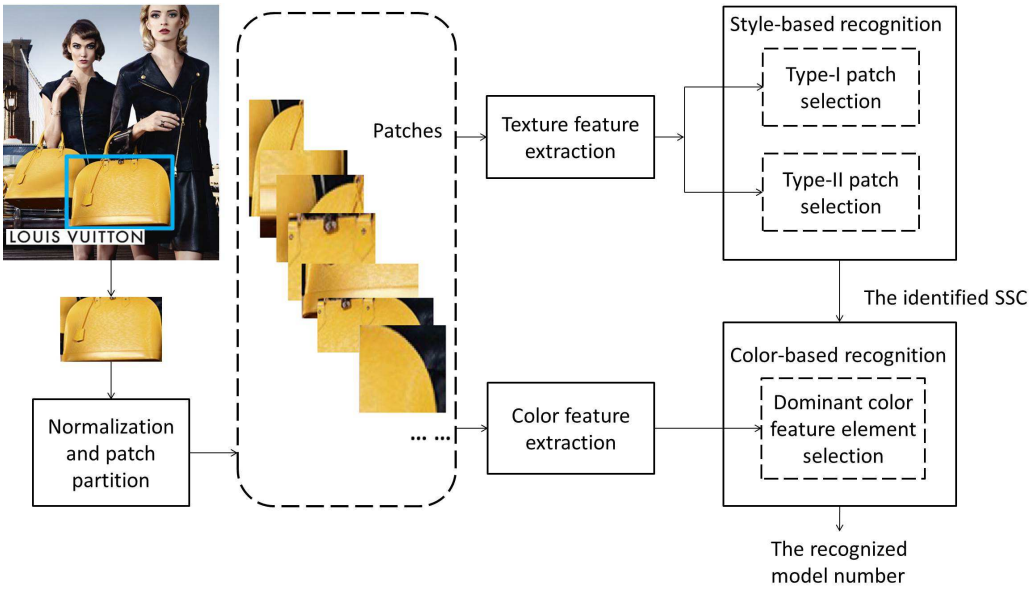


Figure 3.2: Overview of the designed handbag recognition framework.

Users first crop the handbag regions of interest (ROIs) from the images which they captured from fashion magazines or downloaded from someone's blog. Then they upload the handbag ROIs to our system for recognition. Each handbag ROI is rescaled to the uniform scale, and a set of K patches of multi-scales are extracted densely from it (see Section 3.4 for more details about the patch partition). A sequential method is applied to recognize the models of the input handbag ROIs. For recognizing the handbag style, two supervised patch selection strategies are performed to discover the discriminative ones, where a complementary feature is extracted and concatenated with the original feature to form a mid-level representation for handbags. Then a dominant color feature element selection is designed for recognizing the handbag models within each SSC.

3.2.1 Style-Based Recognition

The aim of this section is to identify which SSC a query handbag belongs to. The key problem is to find and make use of the discriminative handbag patches for

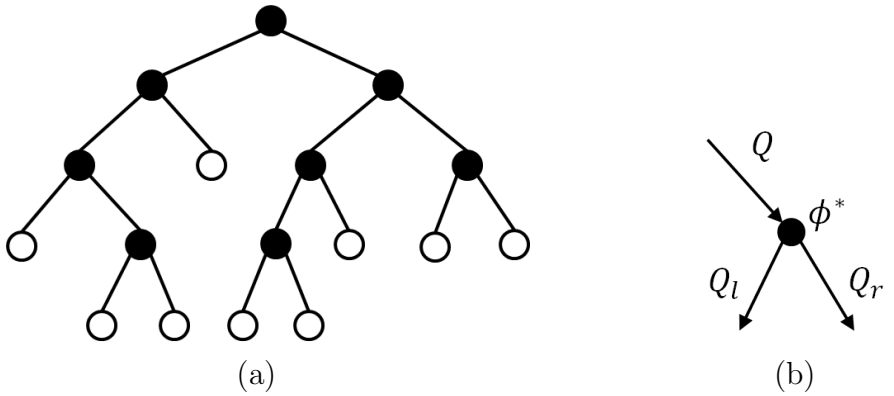


Figure 3.3: Structure of a random decision tree. (a) Each tree consists of branch nodes (filled circle) and leaf nodes (empty circle). (b) For each branch node, a set of input training samples Q are needed to be binary splitted into left and right subsets Q_l and Q_r.

recognition. We propose two patch selection mechanisms to select the discriminative patches w.r.t. one individual SSC and pairwise SSCs.

3.2.1.1 Patch Selection for Individual SSC

The texture difference among handbag SSCs can be very subtle and only some patches matter. It is necessary to find the discriminative patches that can capture the intrinsic characteristics of each SSC. To this end, we propose a random forest-based strategy [40] to measure how discriminative a patch is for identifying handbag styles.

Each random forest tree consists of several branch nodes and leaf nodes, as illustrated in Fig. 3.3(a). In each branch node, the input training samples Q are split into left and right subsets Q_l and Q_r respectively, as shown in Fig. 3.3(b). The binary split is determined by a pair of well chosen parameters $\phi^* = (\theta^*, \tau^*)$, where θ^* is the index of the most discriminative (i.e., θ^{*th}) feature element and τ^* is the corresponding threshold. Usually the quality of the split can be measured by using the reduction of Gini impurity [109] before and after partition. Here, we

follow the same procedure in the standard random forest pipeline [110, 111] for selecting those most discriminative feature elements and storing their indexes. We propose to measure the discriminability of a patch based on its feature elements that are selected among all the nodes.

Let's denote the feature of the k^{th} $(k = 1, 2, \ldots, K)$ partitioned patch for a handbag ROI r as $\mathbf{f}(r(k))$, a $W \times 1$ dimensional vector. By concatenating the features of all the K patches, we obtain a $(K \times W) \times 1$ dimensional vector for each handbag ROI. We then propose to measure the discriminability of the k^{th} patch by

$$d(k) = \sum_{t=1}^{T} \Phi(t, k), \tag{3.1}$$

where T is the number of the branch nodes for all trees and

$$\Phi(t, k) = \begin{cases} 1 & \text{if } \theta^*(t) \in [(k-1)W + 1, kW] \\ 0 & \text{otherwise,} \end{cases} \tag{3.2}$$

where $\theta^*(t)$ refers to the index of the feature element chosen in the t^{th} node. In our implementation, for each splitting node, one feature element is selected by the Gini impurity measure among a randomly chosen subset of features (i.e., $\sqrt{K \times W}$ out of $K \times W$). Such that the training samples reaching this node can be best split. θ^* is the dimension of this selected feature element. The Y most discriminative patches obtained according to Eq. (3.1) (i.e., the patches with the Y largest $d(k)$) are selected for future recognition.

3.2.1.2 Patch Selection for Pairwise SSCs

The aforementioned patch selection considers the overall discrimination of the patches for an individual SSC. However, the patches that best differentiate the

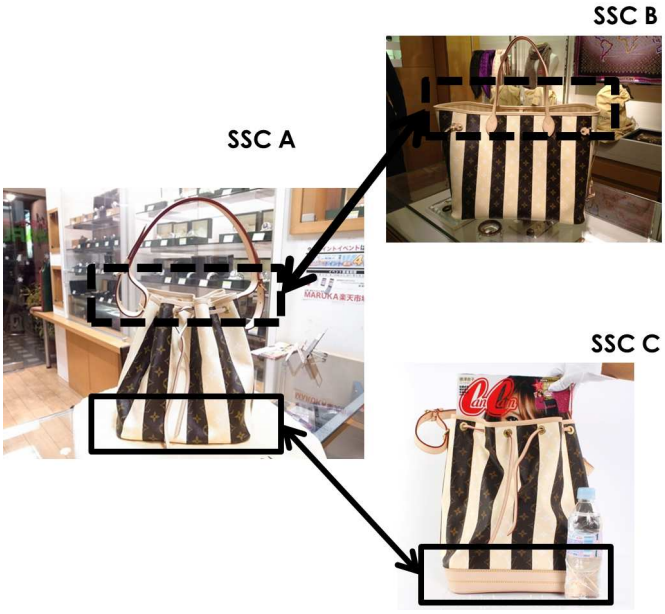


Figure 3.4: Examples for the most discriminative part of pairwise SSCs. The dotted rectangles (upper region) are the most distinctive part for SSC A and SSC B; the rectangles with solid lines (lower region) are the most distinctive part for SSC A and SSC C.

two similar SSCs may not be treated as discriminative patches to classify other SSCs. To enable our style-based recognition to capture the differences between SSCs with very similar styles, we propose to find the most discriminative patch for a pair of SSCs. The location of the most discriminative patches for different pairs of SSCs may vary, as shown in Fig 3.4.

Assume that there are a pair of SSCs used for training: $A = \{a_i\}_{i=1}^{n}$ and $B = \{b_j\}_{j=1}^{m}$, where n and m are the numbers of normalized handbag regions from SSC A and SSC B respectively, and a_i and b_j denote i-th and j-th normalized handbag regions from SSC A and SSC B, respectively. To simplify the symbol, we drop the subscript i, j and define $d_{a,b}(k)$ as the Chi-square distance between the k^{th} $(k = 1, ..., K)$ patches in normalized handbag ROI a and b:

$$d_{a,b}(k) = \chi^2 \left(\mathbf{f}(a(k)), \mathbf{f}(b(k)) \right). \tag{3.3}$$

We then define the discriminability of corresponding k^{th} patch for a pair of SSCs (A, B) as

$$D_{A,B}(k) = \frac{d_{A,B}(k)}{\left(d_{A,A}(k) + d_{B,B}(k)\right)/2}, \tag{3.4}$$

where

$$d_{A,B}(k) = \frac{1}{n \times m} \sum_{\substack{i=1,...,n,\\ j=1,..,m}} \left(d_{a_i,b_j}(k)\right), \tag{3.5}$$

$$d_{A,A}(k) = \frac{1}{n \times (n-1)} \sum_{\substack{i,i'=1,...,n,\\ i \neq i'}} \left(d_{a_i,a_{i'}}(k)\right), \tag{3.6}$$

$$d_{B,B}(k) = \frac{1}{m \times (m-1)} \sum_{\substack{j,j'=1,...,m,\\ m \neq m'}} \left(d_{b_j,b_{j'}}(k)\right). \tag{3.7}$$

For the k^{th} pair, Eq. (3.5) measures an inter-class distance between SSC A and SSC B; Eq. (3.6) and Eq. (3.7) measure the intra-class distances of SSC A and SSC B respectively. Suppose there are in total N SSCs for a certain brand, for i^{th} $(i = 1, ..., C_N^2)$ pair of SSCs, we choose the $Z(i)^{th}$ patch among the K pairs of corresponding patches as the most discriminative patch for future recognition,

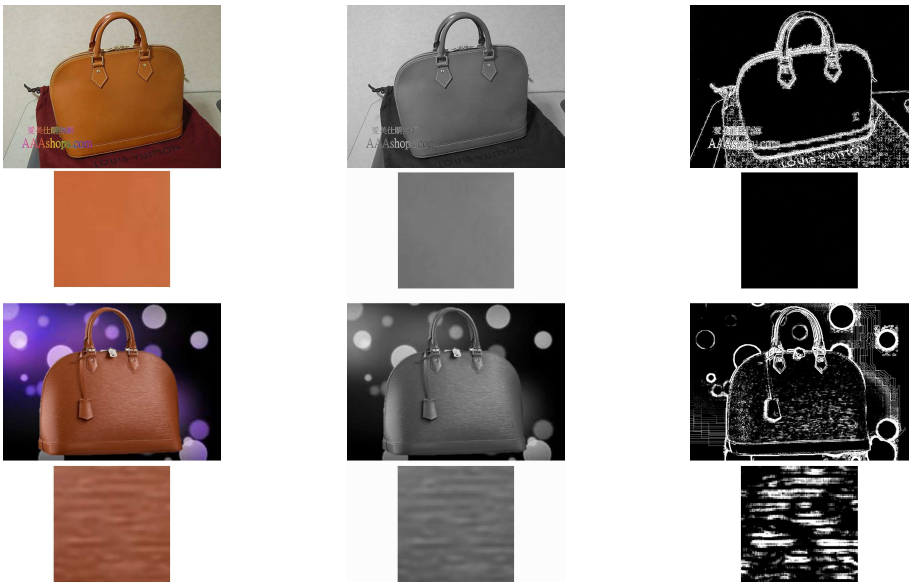
$$Z(i) = \arg\max_k \left\{D_{A,B}(k)\right\}. \tag{3.8}$$

Now that we have selected two types of discriminative patches to facilitate the SSC recognition, we term these two types of discriminative patches as the following.

1. Type-I patch: one of the Y discriminative patches selected for an individual SSC.

2. Type-II patch: the most discriminative $(Z(i)^{th})$ patch for i^{th} pair of SSCs.

Among N SSCs, first of all, we concatenate the features extracted from the Type-I patches and form a feature vector $\mathbf{g}$. To differ the i^{th} pair of SSCs, we train



(a) (b) (c)

Figure 3.5: Two handbags with similar styles. (a) Original handbag images, (b) Gray images, (c) α-images.

a binary classifier $R(i)$ based on the features extracted from the $Z(i)^{th}$ patch of the pair of SSCs, where $Z(i) = 1, 2, ..., K$ is the Type-II patch of this pair of SSCs. Since the number of $R(i)$s is somewhat large (in total C_N^2 binary classifiers), we adopted the fast and effective logistic regression formulation [112] for the training. After each $R(i)$ has been trained, we extract the feature of the $Z(i)^{th}$ patch of each handbag ROI and feed it to $R(i)$ to obtain a score. We then concatenate all the scores (in total C_N^2 scores) and form a feature vector $\mathbf{g}'$. The final feature vector $\mathbf{h}$ is obtained by concatenating $\mathbf{g}$ and $\mathbf{g}'$. Then it is adopted to train a N-class SVM classifier C for SSC recognition.

3.2.2 Complementary Feature Extraction

We can use any of the existing features developed to represent handbag patches, such as SIFT [31], HOG [34], LBP [32], or MR filter sets (texton) [36]. However, we find that the subtle difference among different handbags may not be shown

prominently in the original image due to the lighting conditions or out of focus. Fig. 3.5 shows two different handbags, it is difficult to tell the difference between these two handbags from the original images or gray images (see Fig. 3.5(a)(b), first and third rows). If we look into the zoom-in patches (see Fig. 3.5(a)(b), second and fourth rows), the details of the textures are shown but still not prominent. Therefore, applying the existing features directly on the original image (i.e., the original feature) may not capture sufficient details of the handbag textures.

In this section, we propose to extract the feature from the texture enhanced image, which is termed as the complementary feature [91], to complement the details of the texture that are not captured by the original feature. The enhancement is performed by using Hölder exponent from the multifractal theory [113]. The Hölder exponent is given as follows.

$$\alpha = \lim_{\varepsilon \to 0} \frac{\ln\left(\hbar(S_\varepsilon)\right)}{\ln\left(\varepsilon\right)}, \tag{3.9}$$

where S_ε refers to a local squared region with the length of the side ε, $\hbar(S_\varepsilon)$ is a local regularity measure for S_ε, which can be maximum, minimum, sum, iso measure or absolute difference of the squared region as suggested in [114]. As indicated in [113], Hölder exponent has the ability to enhance the image local structure. It works well even when the differences among local neighborhood are subtle. However, Eq. (3.9) is defined in a continuous space, it is necessary to convert it to discrete domain when applying on images. Then S_ε refers to the squared local image block and ε denotes the number of pixels along one side. According to [113], the value of α is estimated as a slope of linear regression line in discrete space. $\varepsilon = 1, 3, 5, ...$ is the neighborhood size. Slightly different from [113], to efficiently capture the regularity of the local structure for a pixel located at (i, j) in the image, we consider its nearest neighborhood up to $\varepsilon = 3$ (i.e., $\varepsilon = 3$

and 1). Thus the Hölder component for this pixel is computed as

$$\alpha(i,j) = \frac{\ln \hbar(S_3(i,j)) - \ln \hbar(S_1(i,j))}{\ln(3) - \ln(1)}, \tag{3.10}$$

where $S_\varepsilon(i,j)$ refers to the $\varepsilon \times \varepsilon$ block with the center located at (i,j). In our implementation, $\hbar\left(S_\varepsilon(i,j)\right)$ is the maximum pixel value within $S_\varepsilon(i,j)$, i.e.,

$$\hbar(S_\varepsilon(i,j)) = \max_{(k,l)\in S_\varepsilon(i,j)} g(k,l), \tag{3.11}$$

where $g(k,l)$ denotes the value of the pixel located at (k,l).

$\alpha(i,j)$ can be normalized to $\alpha^*(i,j) \in [0,255]$ to form a grayscale image α^*, which is termed as α-image (see Fig. 3.5 (c)). Observe that the details of handbag textures are enhanced in the α-images, which provide more discriminative information to recognize different handbags. After texture enhancement, the complementary feature can be extracted from the α-image of the handbag using existing feature extractors such as SIFT, HOG, etc., which can further improve the discriminative power of mid-level handbag patches. We are the first to incorporate α-image with those feature descriptors to improve the classification performance.

3.2.3 Color-Based Recognition

In this section, we proceed to further recognize the handbag model within each SSC based on the color information only. As discussed before, handbags within one SSC differ from each other just by their colors. Due to the illumination disturbance under real circumstances, it always poses challenges for recognizing the colors. Instead of using 3 primary color components (i.e., RGB) of the image, we adopt the color naming method [115]. Color name learns color from real-world images rather than learning color chips in a laboratory setting. It maps different shades of a color (caused by varying illumination or shadows) into the same color name.

Figure 3.6: Examples of different handbag models in one SSC. The color from certain patches is more discriminative (in dotted boxes).

Figure 3.7: Handbags with the corresponding color histograms. The first row and third row are two handbags with three different images per handbag model. The second row and fourth row are corresponding extracted color histograms.

It can also encode achromatic colors (e.g., black), leading to higher discriminative power [116].

A color naming feature is an 11-dimensional histogram calculated in CIELAB color space. This color histogram indicates the probabilities that an image contains 11 different colors, namely *black*, *blue*, *brown*, *grey*, *green*, *orange*, *pink*, *purple*, *red*, *white* and *yellow*, respectively. However, using an 11-dimensional histogram to differentiate handbag models in one SSC is not discriminative enough for two reasons:

1. Sometimes only the color of some corresponding patches are different, as shown

in Fig. 3.6;

2. Sometimes the color histograms for images belonging to the same handbag vary a lot because of the diverse illumination environments, such as handbags with glossy material, as shown in Fig. 3.7. However, a few color elements may remain consistent with each other (see the third color element shown with horizontal textures in Fig. 3.7).

Therefore, we extract an 11-dimensional color histogram for each of the K patches from a handbag ROI. It would be helpful to find certain color elements extracted from certain patches to facilitate the color recognition, which are termed as the dominant color feature elements.

To describe the color information of a handbag region, we concatenate 11-dimensional color histogram extracted from each handbag patch, and form a $11 \times K$-dimensional color feature. We obtain the dominant color feature elements for a certain SSC by following a similar procedure to the Type-II patch selection described in the previous section. Recalling that a SSC $A = \{a_i\}_{i=1}^{n}$, it can be rewritten as $A = \{H_l\}_{l=1}^{L}$, where L denotes the number of handbag models in A and H_l is the set of the normalized handbag regions belonging to l^{th} handbag model. H_l^* denotes the set of handbag regions which do not belong to l^{th} handbag model. Let $q(a)$ be the mapping function between a normalized handbag region a and its model index, i.e. $q(a) \in \{1, \ldots, L\}$, then we can obtain $H_l = \{a_i | q(a_i) = l\}_{i=1}^{n}$. Let $|\cdot|$ be the number of samples in a set, $h_l^* = \{h_t | t = 1, ..., L, t \neq l\}$, and $\bar{d}_{a_i,a_j}(k)$ denote the Euclidean distance between k^{th} $(k = 1, ..., 11 \times K)$ color dimension of normalized handbag regions a_i and a_j $(a_i, a_j \in A)$. The discriminability of the k^{th} color dimension for SSC A is computed by

$$D_A(k) = \frac{1}{L}\sum_{l=1}^{L}\frac{\bar{d}_{H_l,H_l^*}(k)}{\left(\bar{d}_{H_l,H_l}(k)+\bar{d}_{H_l^*,H_l^*}(k)\right)/2}, \tag{3.12}$$

where

$$\bar{d}_{H_l,H_l^*}(k) = \frac{1}{|H_l|\times|H_l^*|}\sum_{\substack{q(a_i)=l,i=1,\ldots,n,\\ q(a_j)\neq l,j=1,\ldots,n}}\left(\bar{d}_{a_i,a_j}(k)\right), \tag{3.13}$$

$$\bar{d}_{H_l,H_l}(k) = \frac{1}{|H_l|^2-|H_l|}\sum_{\substack{q(a_i)=l,q(a_{i'})=l,\\ i,i'=1,\ldots,n,i\neq i'}}\left(\bar{d}_{a_i,a_i'}(k)\right), \tag{3.14}$$

$$\bar{d}_{H_l^*,H_l^*}(k) = \frac{1}{|H_l^*|^2-|H_l^*|}\sum_{\substack{q(a_i)\neq l,q(a_{i'})\neq l,\\ i,i'=1,\ldots,n,i\neq i'}}\left(\bar{d}_{a_i,a_{i'}}(k)\right). \tag{3.15}$$

For the k^{th} color dimension, Eq. (3.13) measures the distance between handbags which belong to l^{th} model and those belong to other models (i.e., inter-class distance). Eq. (3.14) and Eq. (3.15) are the normalized factors, and they measure the distance of handbags which belong to the l^{th} model and other models respectively (i.e., intra-class distance). We rank each dimension according to its discriminativeness computed by Eq. (3.12) in a descending order. First P dimensional color feature elements with the highest discriminability are chosen as the dominant color feature elements. For handbag model(s) of a SSC, if $L > 1$, we further train an L-class classifier based on the dominant color feature elements. For efficiency, we adopt the fast and effective multi-class softmax regression [112] for the handbag model recognition.

3.3 Dataset Construction

As no existing benchmark is available for branded handbag recognition, we construct a handbag dataset, named as BrandBag. It covers 220 BrandBag-I[2] handbag

[2] http://www.louisvuitton.eu/front/\#/engE1/Collections/Women/Handbags

Figure 3.8: Examples of visually indistinguishable handbags. Handbags with the same appearance but with (a) different sizes, (b) indistinguishable colors and (c) different materials.

models and 181 BrandBag-II[3] handbag models downloaded from Google, Flickr or some shopping websites. Building such datasets costs a lot of human labor due to the following reasons. (1) The image resources are limited for most of the handbag models. (2) Slight texture or color changes of handbags will lead to different handbag models. Handbag images of the same model might appear differently due to serious distortions or variations of illumination.

The dataset construction procedure consists of four key steps. (1) List the target handbag models to collect. (2) Merge handbag models which are visually undistinguishable, such as handbags with the same appearance but different sizes, indistinguishable colors, or different materials (as shown in Fig. 3.8). (3) For

[3] `http://www.coach.com/shop/women-handbags?viewAll=true`

each handbag model, search for handbag images from public websites (Google, Flickr or online shops) by key-words such as the model name, which costs human labor because many attached annotations do not match with the images. (4) Remove handbag images which are noisy, duplicated, heavily occluded, with low quality or in wrong viewpoints, and retain handbag models containing at least 10 images. Eventually, the BrandBag dataset contains 5545 images of 220 BrandBag-I handbag models and 4318 images of 181 BrandBag-II handbag models. According to the style of handbags, BrandBag-I handbag models can be grouped into 125 SSCs and BrandBag-II handbag models have 61 SSCs. Each handbag image in our dataset is annotated manually with a bounding box, which is a rectangular region outside the handbag surface without strap, as shown in Fig. 3.9.

Handbag images in our dataset are mainly in frontal view or with small rotations (i.e., the angle of the rotation is smaller than 30°), which is the best view to visualize handbags. Handbags appearing in these images are without heavy occlusion (i.e., the occluded region is no larger than 15% of the handbag size). The sizes of the images range from 93×99 to 3648×3968, while the sizes of the handbags in images are from 50×96 to 2025×3707. To the best of our knowledge, this is the first handbag dataset constructed for branded handbag recognition.

3.4 Experiments and Discussions

3.4.1 Evaluation of Proposed Method

In this section, we evaluate the performances of the handbag recognition on the branded handbag datasets. It is also worth noting that our method requires users to crop the handbag ROIs from the handbag images, therefore, the background of the image and the position of handbag will not affect the recognition. Methods

Figure 3.9: Examples of handbag images with the associated bounding boxes (marked with yellow rectangles) in our dataset.

Table 3.1: Comparisons of the classification accuracies (%) and feature dimension. CN is short for color naming, CS is short for color selection (color naming feature extraction + dominant color feature element selection) and CF is short for complementary feature.

Methods	Datasets			Feature dimensions	
	BrandBag	BrandBag-I	BrandBag-II	Style	Color
Baseline-I	54.08	67.21	39.00	21,000	
Baseline-II	72.83	81.33	60.62	21,000	297
Type-I + CN	79.59	88.54	67.86	15,000	297
Type-I + Type-II + CN	82.19	90.77	70.91	19,000	297
Yao *et al.* [4] + CN	74.01	80.11	66.74	$\approx 4,086,000$	297
Singh *et al.* [30] + CN	64.03	71.04	39.68	3906	297
Type-I + Type-II + CF + CN	83.46	92.34	71.96	34,000	297
Type-I + Type-II + CF + CS (Proposed)	**84.01**	**92.77**	**72.25**	34,000	230
Yao *et al.* [4]	61.47	76.28	39.56	$\approx 4,086,000$	
CNN [108]	71.27	80.18	60.69	4096	

will be mainly evaluated and compared on BrandBag. We also show some performances on BrandBag-I and BrandBag-II separately. In the dataset, five images per handbag model are randomly chosen for training and the rest are for testing. The training and testing sets are fixed for the rest of the experiments. We adopt cross-validation to obtain the parameters in the proposed framework. In the following discussions, we denote SIFT + LLC to represent LLC feature [33] coded from SIFT descriptor followed by spatial pyramid matching (SPM) [8] and max-pooling [117]. Size of the codebook is 1000 unless otherwise specified. Two baselines considered for comparison are as follows:

- Baseline-I: C-SIFT [118] + LLC.

- Baseline-II: we sequentially recognize the handbag from style to color. SIFT + LLC is used for style recognition, and the color naming [115] is adopted for color recognition.

The classification accuracies of the two baselines on our datasets are shown in Table 3.1. It can be seen that the sequential recognition process can significantly increase the performance for handbag recognition. Next, we are to evaluate the performance of proposed discriminative representations for handbags. Each handbag ROI is rescaled into 255×255 pixels. We observe that handbag designers tend to decorate handbags with patterns or textures in the four corners, middle, upper or lower part, and the two sides. This motivates us to partition each ROI into scales of 85×85, 85×170, 170×85, 85×255 and 255×85 with a step size of 85, which obtains $K = 27$ patches in our experiment.

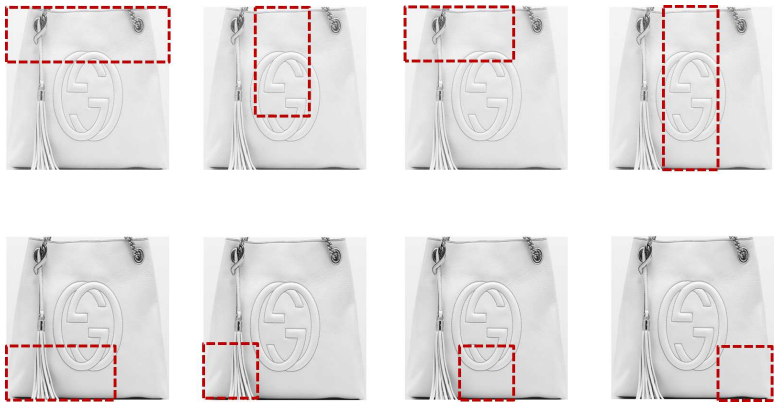


Figure 3.10: Patches arranged by their discriminability in a descending order. We visualize the most four (first row), and the least four (second row) discriminative patches (patches are indicated as dotted boxes).

3.4.1.1 Performance of Type-I Patch Selection

We use the selected Type-I patches instead of the standard SPM patches in baseline-II pipeline and vary the number of selected patches. We select $Y = 15$ most discriminative patches by grid search [119]. The result is shown as Type-I + CN in Table 3.1, which is around 2.1% improvement when compared with selecting all the 27 patches (accuracy = 77.50%). Moreover, we observe that the performance is not sensitive in recognizing handbags. In BrandBag, the accuracies vary from 79.44% to 79.59% for $Y \in [13, 18]$. The values of T are $21,664$, $16,024$, and $37,922$ for BrandBag, BrandBag-I and BrandBag-II, respectively. We also show the feature dimensions for different methods of BrandBag in Table 3.1. It can be seen that with more compact features, the performance of the proposed Type-I + color naming method surpasses the baseline.

Fig. 3.10 visualizes the four most discriminative patches as well as the four least discriminative patches according to Eq. (3.1) in our BrandBag dataset. The ranking shows that the discriminative patches locate on the upper and middle part of the handbags.

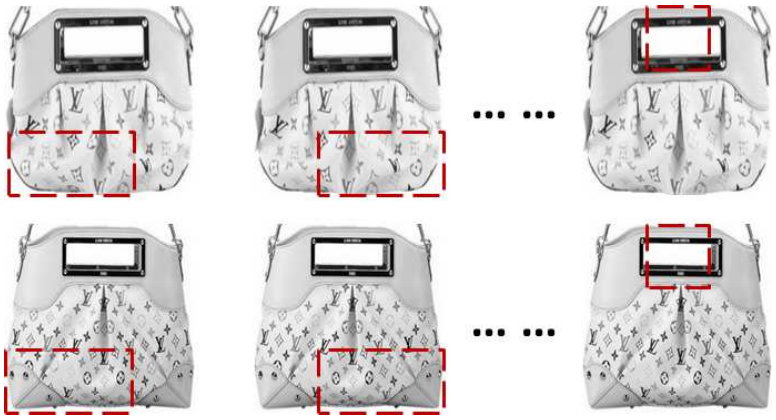


Figure 3.11: An example of learned discriminative patches for two SSCs (shown in two rows respectively). We rank the discriminability of patches in a descending order, and visualize the two most discriminative patches (in dotted boxes) as well as the least discriminative patch.

Table 3.2: Variation with selected number of pairwise classifiers for handbag recognition.

Number of selected $R(i)$	Accuracy(%)
4000	82.19
8000	82.21
12000	82.24
16000	82.20

3.4.1.2 Performance of Type-II Patch Selection

Fig. 3.11 shows the ranking of discriminability of patches for pairwise SSCs in a descending order of BrandBag. Top rank indicates that the diversity of bottom corner is important for distinguishing these two SSCs.

As discussed in Section 3.2.1.2, among N SSCs, C_N^2 binary classifiers will be built in this module to construct a feature $\mathbf{g}'$. There are 186 SSCs in our dataset, yielding $\begin{pmatrix} 186 \\ 2 \end{pmatrix} = 17205$ dimensions for $\mathbf{g}'$. However, it is not necessary to use all the 17205 elements in $\mathbf{g}'$. Table 3.2 shows the performance of handbag recognition

with $\mathbf{g}'$ by using randomly selected number of pairwise classifiers (i.e., $R(i)$) on top of the Type-I patch selection with $Y = 15$. We denote the number of selected pairwise classifiers as E and we are able to gain a good performance when $E = 4000$, which achieves 82.19% in accuracy, and set as a default value. With the increasing number of pairwise classifiers, the redundancy in our features increases as well, which may result in accuracy decrease in Table 3.2. This is around 2.6% improvement compared with only using the Type-I patches with $Y = 15$ (accuracy = 79.59%). For BrandBag-I and BrandBag-II, the improvements are 2.23% and 3.05%, respectively.

3.4.1.3 Comparisons of Discriminative Patch Discovery

We replace the proposed patch selection with the leading and publicly available method [4] for recognizing fine-grained objects (see Yao *et al.* [4] + CN in Table 3.1) and discriminative patch discovery method [30] (see Singh *et al.* [30] + CN in Table 3.1). We adopt the default settings in their source codes. Yao's method randomly samples patches from images, and the average feature dimension per image is $4,086,000$ in BrandBag. For Singh's work, we discover discriminative patches on a per-category basis, and aggregate top discovered patches of each SSC into an object bank representation [120]. It ends up with a 3906 dimensional feature vector for each image. Our patch selection for the task achieves over 8% improvement in accuracy for BrandBag, 10% for BrandBag-I and 4% for BrandBag-II.

3.4.1.4 Performance of Complementary Feature Extraction

We use the term Hölder-SIFT to denote the complementary feature of SIFT. We concatenate SIFT + LLC with Hölder-SIFT + LLC to represent handbag patches

in Type-I + Type-II + CN, which achieves 83.46% in accuracy for BrandBag, 92.34% for BrandBag-I and 71.96% for BrandBag-II (see Type-I + Type-II + complementary feature + color naming in Table 3.1). We observe the improvement of embedding complementary feature into our framework, which usually boosts the performance by 1.5% in recognition accuracy.

3.4.1.5 Performance of Dominant Color Feature Element Selection

As mentioned in Section 3.2.3, we choose the P most discriminative color elements (out of $11 \times 27 = 297$ dimensional feature extracted from all patches) for the color-based recognition. We are able to achieve the performance of branded handbag recognition (on top of the Type-I and Type-II patch selection, based on grid search) at $P = 230$ with accuracy of 84.01% for BrandBag, 92.77% for BrandBag-I and 72.25% for BrandBag-II. In our dataset, the number of handbags in each SSC, denoted by L, is relatively small , where $L \in [1, 13]$ and the average value is 2.16. This may be one of the reasons why the improvement is little compared with using all the color feature elements (e.g., 83.46% of BrandBag in accuracy without dominant color feature selection).

Finally, we compare our method with that proposed by Yao *et al.* [4] on handbag recognition, i.e., we directly adopt Yao's method for recognizing different handbag models, denoted as Yao *et al.* [4] in Table 3.1. As Convolutional Neural Network (CNN) has been shown recently to be effective for many image recognition tasks, we adopt the ImageNet pre-trained model [108] and fine-tune it on our handbag data. We start the training with a fixed learning rate and decrease it by a factor of 10 after the training error stops reducing, and the result is reported in Table 3.1. Our proposed method achieves over 13% improvement in classification accuracy.

3.4.2 Discussion on Image Size

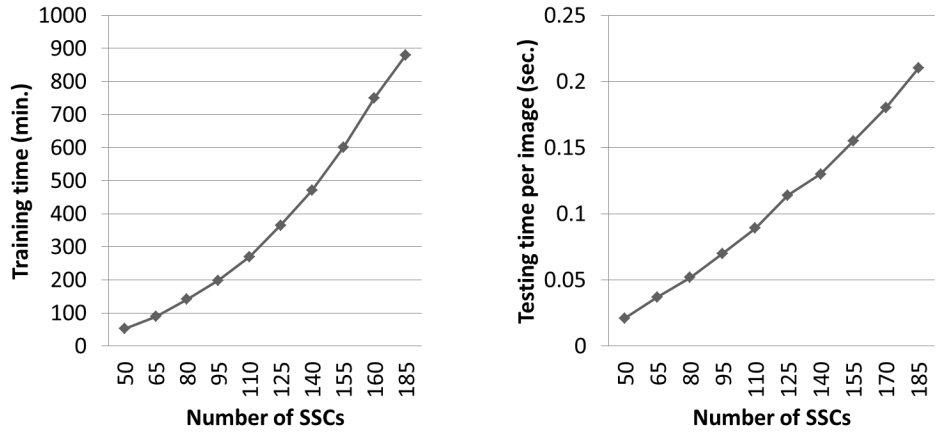
We test the effects of handbag size on the recognition accuracy. Take BrandBag-I dataset as an example, we rescale each handbag image in the testing set of Brandbag-I into half of its original size, w.r.t height (i.e., $H' = 1/2 \times H$, where H indicates the original height and H' indicates the rescaled height) and width (i.e., $W' = 1/2 \times W$, where W indicates the original width and W' indicates the rescaled width). The bounding box annotation is also changed accordingly. Then we follow the same procedure as that has been used in Type-I + Type-II + complementary feature + color selection (Proposed). More specifically, we rescale each handbag ROI to the same uniform scale (i.e., 255×255), densely extract K patches of multi-scales from the rescaled ROI and recognize the models of the input handbag ROIs by using the sequential method and already trained classifiers. We obtain the accuracy of 93.58% for our proposed method. After resizing the image, the accuracy does not drop, compared with the reported accuracy in Table 3.1 (92.77%). Therefore, the recognition accuracy is not sensitive to the size of handbag images in our dataset.

3.4.3 Discussion on the Number of Training Images

In order to evaluate how the number of training images influences the performance, take BrandBag-I dataset as an example, we randomly select 8 images per handbag model for training, and the rest are for testing. This leads to 94.46% in accuracy. It can be seen that we can obtain better performance with more training images.

3.4.4 Computational Complexity

We report the time complexity of training and testing on BrandBag dataset for proposed method. Experiment is conducted on Matlab R2012b, in a desktop



(a) (b)

Figure 3.12: Training and testing time based on different number of SSCs: (a) overall training time (min.), and (b) testing time for each query handbag ROI (sec.).

of Intel(R) Core(TM) i5-4570 3.20GHz CPU, and 32.0GB RAM. Fig. 3.12 (a) gives the curve for measuring the total training time (in min.) changing with the increasing number of SSCs. Fig.3.12 (b) plots the testing time (in sec.) for each query.

For Type-II patch selection, we adopt all the C_N^2 binary classifiers to keep the measurement fair and simple. However, it is not necessary to train all the binary classifiers and the speed would become much faster if we reduce the number of binary classifiers to be trained. When $K = 186$, and we do not adopt any binary classifier, the training time will reduce to 51.40 seconds. This will sacrifice approximately 2.5% in accuracy. Moreover, we compute the overall training and testing time sequentially for Type-I, Type-II patch selection and color element selection modules. To speed up, these modules can be computed in parallel. Based on Fig. 3.12 (b), our handbag search engine is able to be applied in real-time.

3.4.5 Discussions

Our propose method requires to group the handbags into SSCs to facilitate the style-based recognition as well as the color-based recognition. In order to identify new handbags, we have to group them into the existing or new SSCs. A straightforward solution is to let experts manually label which SSC the new handbag should be assigned to. To reduce the human labor, we can use the style-based recognition to identify which SSC this handbag belongs to or provide several candidate SSCs which this handbag likely belongs to. This will help the experts to largely narrow down the range of choices. If the new handbag belongs to an existing SSC, we only need to train the L-class classifier for color-based recognition element within this SSC. If the new handbag forms a new SSC, then most of the additional time cost will depend on training the binary classifiers ($R(i)$) between the new SSC and the existing SSCs.

The performance of our method is evaluated for a combination of two brands. As our method is not designed for some specific brands only, once there are sufficient data for training, we can directly apply it to even more brands or a combination of more different brands.

3.4.6 Additional Experiments on Complementary Feature

We test the complementary feature based on the LLC framework [33] on our BrandBag dataset, other five widely used image classification datasets: Caltech-101 [121], Caltech-256 [122], MIT 67 indoor [123], Scene 15 [8], UIUC sports [124] and some fine-grained object datasets: Oxford flower [5], Stanford dog [6] and UCSD bird [125]. We also test the complementary feature based on an existing fine-grained object recognition method for bird recognition [4]. Gray scale information is used for all these datasets.

Table 3.3: Accuracy (%) of LLC framework on handbag recognition using different types of features.

Feature	Acc.	Feature	Acc.	Feature	Acc.	Feature	Acc.
SIFT	54.81	Hölder-SIFT	52.11	SIFT & ST-SIFT	54.95	SIFT & Hölder-SIFT	**57.44**
HOG	48.69	Hölder-HOG	46.33	HOG & ST-HOG	48.08	HOG & Hölder-HOG	**51.55**
LBP	37.65	Hölder-LBP	35.01	LBP & ST-LBP	37.46	LBP & Hölder-LBP	**40.63**
Texton	53.39	Hölder-Texton	50.25	Texton & ST-Texton	53.29	Texton & Hölder-Texton	**57.56**

For handbag recognition, different types of features are adopted during the recognition, i.e., the original feature, the complementary feature, and the feature concatenated by the original and the corresponding complementary feature. We apply several popular features including SIFT [31], HOG [34], LBP [32] and Texton [36] as the original features. For simplicity, the corresponding complementary features are termed as Hölder-SIFT, Hölder-HOG, Hölder-LBP and Hölder-Texton, respectively. Table 3.3 summarizes the recognition accuracies when using single feature and it shows the comparison results of accuracies. When using the original features, the accuracies vary from 37.65% (LBP) to 54.81% (SIFT). The concatenation of the original feature and the complementary feature is denoted as SIFT & Hölder-SIFT, HOG & Hölder-HOG, etc. We observe that most of the complementary features perform worse than the original features because they enhance high-frequent information which is sensitive to noise. However, concatenating the original feature and the corresponding complementary feature can get a better performance (over 2.63% improvement in accuracy). To demonstrate the effectiveness of extracting features from our texture enhanced image, we compare it

with sketch token [109], a local edge-based feature. Likewise, we denote the corresponding features extracted from sketch tokens by ST-SIFT, ST-HOG, ST-LBP and ST-Texton. The 5^{th} and 6^{th} columns of Table 3.3 show the performance of concatenating the edge-based feature and the original feature, which does not help in boosting the accuracy.

We test on five widely used image classification dataset with the most standard settings. On Caltech-101, we randomly select 15 and 30 images per class for training, and no more than 50 images per class for testing. On Caltech-256, for each class, 15, 30 and 45 images are randomly selected for training, while the remaining for testing. On MIT 67 indoor, we follow the original splits [123], which uses around 80 and 20 images per class for training and testing, respectively. On Scene 15, 100 images per class are randomly selected for training and the rest are for testing. On UIUC sports, we randomly select 70 and 60 images per class for training and testing, respectively. For the three fine-grained object datasets, we follow their standard data splitting strategies. On Oxford flower, whole images are applied for training and testing. For Stanford dog, only foreground images are adopted. We follow the data preparation of UCSD bird in the work of Yao *et al.*[4]. We trained codebooks with 4096 bases for Caltech-256 as in [33] and 2048 for other datasets. For all the SIFT features extraction and LLC coding, we adopt the default settings in the source code [33]. As shown in Table 3.4, incorporating the complementary feature on general object recognition helps in increasing the performance by over 3% in classification accuracy. Other than accuracy, we also provide the percentage increment of computation time for each dataset in the last column. Note that the increment of computation time of our algorithm is trivial. It costs average 1.1 second per image on a normal computer.

For bird recognition proposed by Yao *et al.* [4], we adopt the default settings in

Table 3.4: Mean classification accuracy (%) and percentage increment of computation time (%) for image classification (with/without the complementary feature).

Datasets	Feature		Percentage increment of
(Number of training images/class)	SIFT	SIFT & Hölder-SIFT	computation time
Caltech-101 (15)	63.82	68.43	155.04
Caltech-101 (30)	71.93	76.27	134.29
Caltech-256 (15)	30.29	33.78	159.93
Caltech-256 (30)	36.77	40.63	165.75
Caltech-256 (45)	40.13	44.09	151.31
MIT 67 indoor (80)	41.51	46.57	160.24
Scene 15 (100)	81.65	84.66	153.93
UIUC sport (70)	81.50	87.71	156.27
Oxford flower (10)	42.38	48.88	150.53
Stanford dog (105)	20.76	24.08	149.34
UCSD bird (15)	12.71	15.73	150.65

Table 3.5: Mean classification accuracy (%) of with/without complementary feature on the existing fine-grained recognition method [4] for bird recognition

Number of decision trees	Feature	
	SIFT	SIFT & Hölder-SIFT
50	11.97	13.74
100	13.49	15.97
150	14.55	16.83
200	15.22	17.64
250	15.43	18.00
300	15.68	18.32

the source code [126], and change the number of trees and size of images given in [4]. The results in Table 3.5 show the improvement of incorporating complementary feature.

3.5 Summary

In this chapter, a novel method is presented to recognize handbag models. Our study focuses on the discriminative representations for handbags. For style-based recognition, two different mid-level handbag patch selection algorithms are proposed to find the discriminative handbag patches, and a low-level complementary feature is designed to capture the fine details of the handbag patches. For color-based recognition, the dominant color naming features are selected and used to represent the color of the handbag. The experiments show that our method performs better than existing fine-grained object recognition methods on recognizing handbags.

Chapter 4

DeepBag: Feature Selective and Joint Classification-Regression for Handbag Recognition

With the development of GPU device and the rise of deep learning methods in recent years, many computer vision applications such as object detection or image classification have gained promising results. As reviewed in Chapter 2, Convolutional Neural Network (CNN) is a variant in deep learning for image classification.

In this chapter, we are motivated to propose a novel handbag recognition algorithm based on CNN [1]. Concretely, we propose a new CNN model, called Feature Selective joint Classification-Regression CNN (FSCR-CNN). Its advantages lie in two folds. (1) It alleviates the illumination changes. (2) It leads to a better handbag classifier to handle large inter-class similarity. We evaluate the performance of FSCR-CNN on our branded handbag dataset introduced in Chapter 3. The experiments show that it outperforms those CNN models on handbag recognition. We also apply FSCR-CNN to other fine-grained object recognition problems based on state-of-the-art CNN architectures, and achieve over 5% improvement in accuracy. Since both Chapter 3 and this chapter deal with handbag recognition, the

[1] Part of this chapter was presented in [127]

Figure 4.1: Illustrations of the main difficulties in handbag recognition due to (a) illumination changes and (b) inter-class similarity. The models of handbags in each row are the same in (a), while similar handbags are enclosed in the same box in (b).

comparisons of methods introduced in these two chapters will be analyzed in the experiment.

4.1 Introduction

As a fashion recognition problem, handbag recognition has a large market demand. New models of handbags appear in the market in a very fast pace. We can see handbag images everywhere, such like fashion magazines, billboards, websites and blogs. To process such large amount of handbag images is the inevitable trend of development. It is nontrivial, which delivers the demand for big data analysis. Under this requirement, in this chapter, we investigate deep learning based

handbag recognition, which has the potential to handle large scale data.

As introduced in Chapter 2, handbag recognition belongs to the category of fine-grained object recognition, which is a challenging problem even for human beings [128]. The main challenges of handbag recognition have been summarized in Chapter 3. We reemphasize them here, which are slightly different from Chapter 3 as the concepts like "style" or "SSC" are not used in this chapter. (1) Illumination changes: some handbags differ with each other only by color (color sensitive), however, the illumination changes enlarge the intra-class color variance (see Fig. 4.1 (a)). (2) Inter-class similarity: the appearances of some handbags may be very similar (see Fig. 4.1 (b)), which raises difficulties to learn a proper classifier to differentiate these visually similar models.

Based on our study, we find that previous CNN models [1, 43, 44, 47] or CNN-based methods for fine-grained object recognition [18, 21, 53] do not provide discriminative color information during training. Moreover, CNN models only consider the hard label (i.e., the ground truth class label) to train a multi-class classifier. This is not sufficient for training a reliable classifier especially for visually similar classes. To handle the difficulties in handbag recognition, we introduce two novel techniques in CNN training. We explicitly consider (1) incorporating discriminative color information to classify color sensitive objects, and (2) assigning a distribution measuring how similar this class is to other classes to differentiate similar classes.

Concretely, we propose a novel CNN model to recognize the handbag model for a given input handbag image. The proposed model, named FSCR-CNN, has the following two innovations.

1. Feature Selective CNN (FS-CNN): we learn a regression function to map the

first fully-connected feature to the color feature via random forest. Then, we measure the color-discriminability of each element of the fully-connected feature based on the regression function. Only those color-nondiscriminative ones will be forwarded and back-propagated.

2. Joint Classification-Regression CNN (CR-CNN): we propose a novel loss function by considering both the hard labels and soft labels of the training data for CNN fine tuning. For each training sample, the hard label means its ground truth class label, while the soft label refers to a distribution that measures the similarities between its ground truth class and all the classes.

Additionally, we also build up an end-to-end handbag recognition framework, given a handbag image. In this framework we first propose to localize and extract a set of handbag regions (proposals) by exploring the symmetry property of the handbag (Sec. 4.3.1). These extracted proposals are then fed into the CNN detection model and the FSCR-CNN (Sec. 4.2) classification model separately. The detection scores and classification scores are combined by employing a conditional probability model (Sec. 4.3.3). Eventually, we recognize the query handbag image according to the highest combined score.

The major contributions of this work can be summarized as follows:

1. We propose a feature selection strategy to improve the discriminability of the learned CNN by optimizing the color-nondiscriminative features in handbag recognition.

2. We propose a novel loss function for CNN by taking both the hard label and the soft label into consideration, so as to facilitate the classifier modeling for visually similar objects. Such a loss function can be adopted on different CNN architectures, with over 7% improvement in accuracy for handbag recognition.

3. The proposed FSCR-CNN can be generalized to other image-based fine-grained object recognition problems.

Next, we present our FSCR-CNN model first, then we introduce the end-to-end handbag recognition framework.

4.2 FSCR-CNN Classification Model

Deep CNN model is shown to be a powerful image descriptor or classifier [1, 129]. However, for fine-grained datasets which only have limited resources, CNN suffers from over-fitting. Therefore, for all CNNs trained in this chapter, we adopt the ImageNet pre-trained model, and fine-tune it accordingly. More specifically, we replace the last fully-connected layer (this layer's outputs are the 1000 classes scores for 1000-class image classification) by a new fully-connected layer whose number of outputs equals to the number of handbag categories in the dataset. We initialize this layer with weights from scratch. We fine-tune the weights of the pre-trained network by continuing the backpropagation. ImageNet [22] organizes objects according to the WordNet [130] hierarchy and each node is depicted by a large amount of images. It contains the subset of bags categorized by the shoulder bag, evening bag, clutch, reticule and etui. Our second strategy which avoids over-fitting is to conduct the data augmentation during training [131], leading to larger diversity of the class.

4.2.1 Feature Selective CNN Architecture (FS-CNN)

In deep CNNs, after the first convolutional layer, RGB channels are all mixed to be fed into the consecutive layers. Such CNN models may not be good at dealing with illumination changes (see Fig. 4.1 (a)). To address this problem, we introduce a feature selection strategy into CNN to help learn features which can better describe

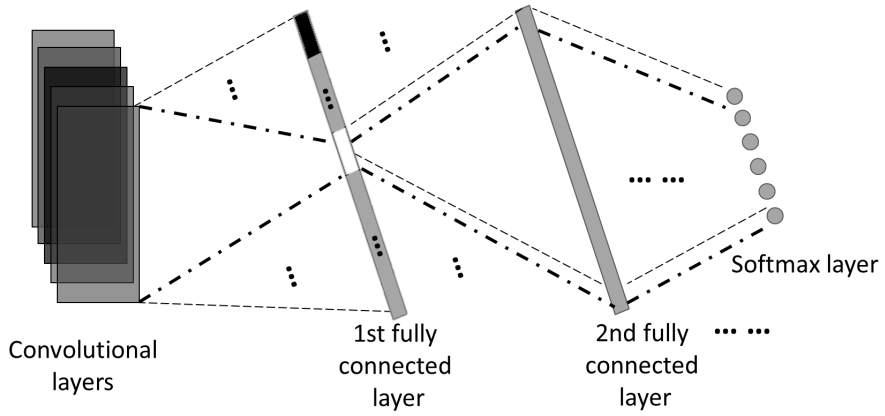


Figure 4.2: Illustration of the FS-CNN. The feature selection is applied on the first fully connected layer, where the black part indicates the color-discriminative feature elements, and the white part indicates the color-nondiscriminative feature elements. The dashed line indicates forward pass and dotted dashed line indicates the backpropagation.

color information. The proposed FS-CNN is shown in Fig. 4.2. Based on the proposed feature selection, the color-discriminative features outputted from the first fully connected layer fc_1 remain unchanged and the color-nondiscriminative features participate in the forward pass and the backpropagation.

We choose color-nondiscriminative features to forward and back-propagate because these features are not informative for the color, which may result in an unsatisfactory classification result. Their associated neurons are required to be further optimized based on the corresponding classification error. In such a way, the whole network is more capable to do the classification. To select the color-nondiscriminative features, we propose a random forest [132] based feature selection procedure. As introduced in Chapter 3, random forest is an ensemble of randomized decision trees, which is shown to perform well for multi-class classifications in many tasks [4, 133, 134]. Each random forest tree consists of several branch nodes and leaf nodes. Each branch node selects a feature dimension which is discriminative for the classification or regression. According to [132], a certain

feature can regress another feature using random forest. Here we adopt such procedure to regress the color feature of each training image using CNN fc_1 feature. In our implementation, the color feature is computed by employing color naming method [115]. During such regression process, each node of the random forest tree will select the most color discriminative dimension of fc_1 feature. We propose to measure the color discriminability of the i^{th} dimension of fc_1 feature by

$$d(i) = \sum_{k=1}^{Y} \Phi(k, i), \tag{4.1}$$

where Y is the number of the branch nodes for all trees and

$$\Phi(k, i) = \begin{cases} 1 & \text{if } \vartheta(k) = i \\ 0 & \text{otherwise,} \end{cases} \tag{4.2}$$

where $\vartheta(k)$ refers to the index of the feature element chosen in the k^{th} node. Among H-dimensional fc_1 feature elements, βH $(0 < \beta < 1)$ most non-discriminative feature elements are selected according to the color discriminability.

4.2.2 Joint Classification-Regression CNN Model (CR-CNN)

Handbag recognition is a multi-class classification problem. A straightforward way to address it is to train a hard label multi-class classifier, such as the softmax classifier. However, some handbags are extremely similar, as shown in Fig. 4.1 (b), even human beings have difficulties to distinguish them. In other words, given a single hard label of a handbag class, it's difficult to train a reliable classifier to distinguish it from its visually similar classes. This is because the penalties for misclassifications to its visually similar classes and dissimilar ones are equal. Therefore, a better way is to assign an additional soft label to each handbag class, which is a distribution to measure how similar this class is to all the classes. By

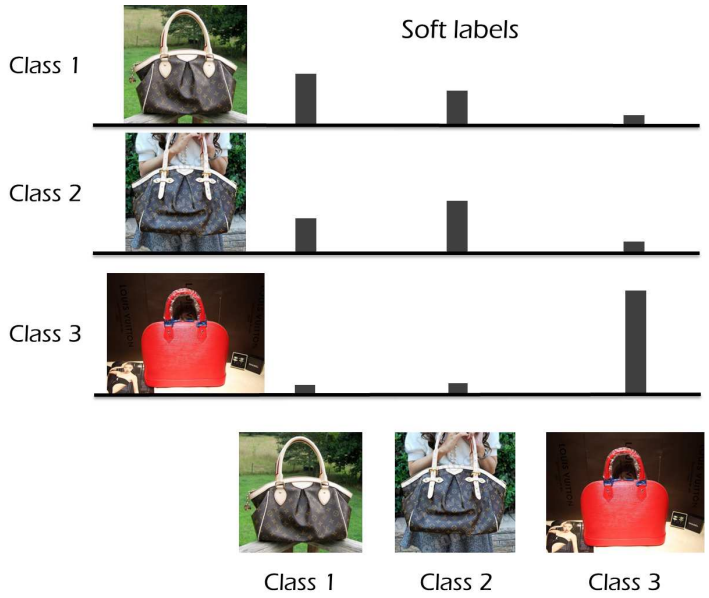


Figure 4.3: Examples of soft labels in a 3-class dataset. The lengths of the histograms measure the similarities between classes. The summation of histograms along any column or any row is 1.

using the soft label, the penalties for misclassifying a handbag to its visually similar classes are less than those to the dissimilar classes. Researches have been done to show that soft label is helpful for some computer vision tasks [135, 136]. Here we propose to take advantages of both hard label and soft label for the CNN training procedure. Fig. 4.3 illustrates the soft labels of a 3-class dataset, each of which can be represented as a 3-dimensional vector.

To assign an additional soft label to each handbag class, we can adopt a confusion matrix based on the classification performance on a validation set as indicated in [137, 138]. However, the confusion matrix measures how easy it is to discriminate between different classes, which may not have a good measure for the similarities between classes. Therefore, in order to establish the soft labels, we propose to use learned CNN features of the training data to measure the similarities between classes. As these features are learned together with classifiers, they are descriptive

and distinctive. The first fully-connected layer (i.e., fc_1) features of the training data are extracted and processed as follows:

1. Compute the mean of fc_1 features among all the training samples within the same class (each class has a corresponding mean fc_1 feature).

2. Obtain a distance matrix $\mathbf{D}$ with each element as

$$D(i,j) = \chi^2(\bar{\mathbf{v}}_i, \bar{\mathbf{v}}_j), \tag{4.3}$$

   where $\bar{\mathbf{v}}_i$ and $\bar{\mathbf{v}}_j$ indicate the mean fc_1 features of class i and j, respectively, $\chi^2(,)$ refers to the Chi-square distance of the mean features.

3. Normalize $D(i,j)$ to $D^*(i,j) \in [0,1]$.

4. Compute a matrix $\mathbf{F}$ with entry

$$F(i,j) = \frac{1 - D^*(i,j)}{\sum_j (1 - D^*(i,j))}, \tag{4.4}$$

   which is the similarity measure of classes i and j.

5. The soft label for class i can therefore be assigned as $\mathbf{s}(i) = (F(i,1), ...F(i,K))$, where K denotes the number of classes being trained.

Now with the learned soft labels, we propose a joint classification-regression loss to learn the network. Given a handbag training dataset with N foreground handbag images $x^{(i)}$ (belonging to K handbag class) with the labels $y^{(i)} \in \{1, 2, ..., K\}$, where $i = 1, ..., N$, and let $a_j^{(i)} (j = 1, ..., K)$ be the output of the last inner-product layer for $x^{(i)}$, the proposed joint loss function is defined as

$$J = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} \left(\mathbf{1}(y^{(i)} = j) \log p_j^{(i)} - \lambda \left(\log F(y^{(i)}, j) - \log p_j^{(i)} \right)^2 \right), \tag{4.5}$$

where

$$p_j^{(i)} = \frac{\exp a_j^{(i)}}{\sum_{l=1}^{K} \exp a_l^{(i)}}, \tag{4.6}$$

$\mathbf{1}(\cdot)$ is the indicator function, and λ is a tradeoff parameter which balances the two loss terms. The first term is the standard softmax loss which penalizes the classification error for each class equally. The second term is the regression squared loss term, which penalizes the difference between the predicted scores of $x^{(i)}$ (i.e., $p_j^{(i)}$) and the soft labels $\mathbf{s}(y^{(i)})$. The second term is learned jointly with the first term, which also acts as a regularizer for the first term.

In order to employ stochastic gradient descend on our proposed loss function, we apply the back-propagation based on the partial derivatives of the new loss with respect to the output of the last inner product layer $a_j^{(i)}$. The partial derivatives are given as follows

$$\frac{\partial J}{\partial a_j^{(i)}} = -\frac{1}{N}\Bigg[\mathbf{1}(y^{(i)} = j) - p_j^{(i)} - 2\lambda\Big(\sum_{l=1}^{K}(\log F(y^{(i)}, l) - \log p_l^{(i)})p_j^{(i)} - \log F(y^{(i)}, j) + \log p_j^{(i)}\Big)\Bigg]. \tag{4.7}$$

4.3 End-to-End Handbag Recognition Framework

Fig. 4.4 shows our proposed handbag recognition framework. Given an input handbag image, we localize a set of handbag regions by exploring the symmetry property of handbags. Two different deep CNN models are trained, the CNN detection model and the FSCR-CNN classification model, to predict the foreground detection score and classification score of each proposal, respectively. Finally, conditioned on the scores of the CNN detection model, a probability model is employed to refine the classification scores of FSCR-CNN model. The model (class) of the input handbag is predicted based on the refined classification scores.

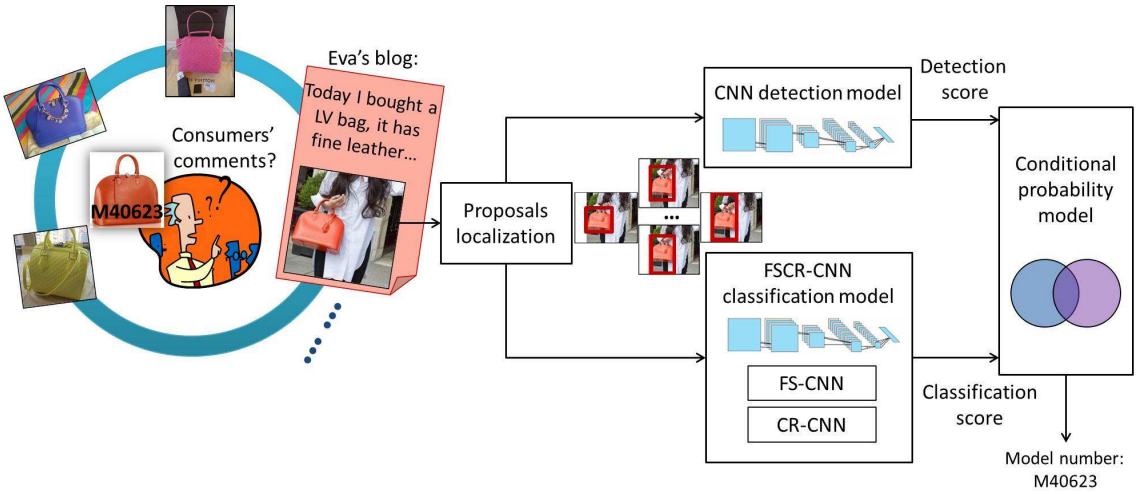
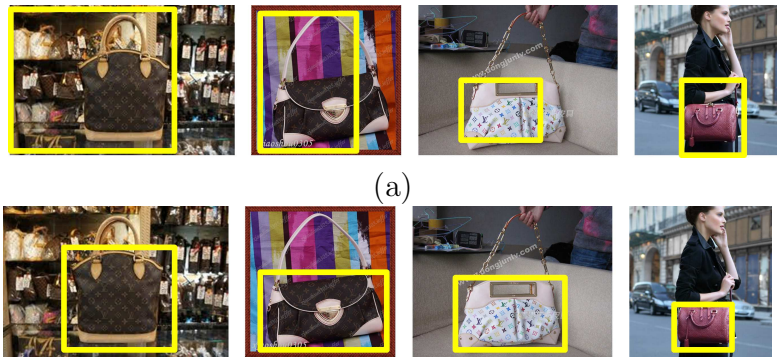


Figure 4.4: Overview of the proposed handbag recognition framework. Given a query (handbag image), a set of proposals are localized and extracted, which are further fed into the CNN detection model and the FSCR-CNN classification model. Eventually, the conditional probability model recognizes the handbag model by combining the classification scores and the detection scores.

4.3.1 Symmetry-Based Proposals Localization

Object proposal indicates a candidate bounding box covering an object in the image [3]. Using object proposals increases the computational efficiency for object detection. Recent works include objectness cues [139], selective search [140], BING with high computation [141] and edge box [3]. Edge box method [3] computes how likely a bounding box contains an object by calculating the number of contours wholly contained in the box, which is suitable for localizing handbag proposals. The shape of handbags is rectangular-like, which satisfies the assumption of the edge boxes, i.e., their contours are more likely to be wholly enclosed or fitted by a box.

However, edge box method is designed for general objects, which does not consider the specific property of handbags. Thus, the returned top proposals by edge box method sometimes cannot accurately enclose a handbag region, which



(a)

(b)

Figure 4.5: Top ranked handbag proposals (enclosed in yellow boxes) by (a) edge box method [3] and (b) proposed method.

cover parts of the region, or parts of the background instead, as shown in Fig. 4.5 (a). This problem can be alleviated by utilizing the observation that handbags are often shown in symmetry.

We follow the notation in [3]. For the computed edge map of an input image I, any of the pixel is defined as e, represented as a complex number, with magnitude m_e and orientation θ_e. The orientation is unsigned, from 0 to π. Then some candidate bounding boxes are computed on the edge map based on a sliding window search. Each bounding box b localizes a corresponding object proposal, the score of which is calculated by: [3]

$$h_b = h_{1b} - h_{2b}, \tag{4.8}$$

where the first term h_{1b} computes the score of whether a set of edge groups is wholly contained in box b, the second term h_{2b} computes the edge magnitudes from a smaller box centered in b, and the subtraction is because those edges in the center of the box are less important.

Next, we propose to compute a symmetry score for the proposal enclosed by box b, which is to measure how symmetric the proposal is. The feature extraction

procedure below for computing symmetry score is shown in Fig. 4.6:

1. Quantize m_e and θ_e into 10 and 6 bins that are uniform in space, respectively.

2. Decompose b into 2×4 spatial cells (4 cells on the right and 4 cells on the left). Each cell is represented by a 6×10-bin 2-dimensional histogram.

3. Normalize the 2-dimensional histogram by the number of edge pixels inside each cell, which is further pulled into a vector to represent the cell.

4. Use the same sequence to concatenate the vector of each cell on the right, denoted as q_r and left, denoted as q_l, respectively.

The symmetry score s_b of the proposal is computed by

$$s_b = -\chi^2(q_r, q_l), \tag{4.9}$$

where $\chi^2(q_r, q_l)$ indicates the Chi-square distance between q_r and q_l. Note that the gradient magnitude and orientation are quantized into 10 and 6 bins empirically. Inspired by the setting of the well-known HOG [34], a 2-D histogram is used to represent a cell. The final proposal score of b is

$$h_b^* = h_{1b} - h_{2b} + \gamma s_b, \tag{4.10}$$

where $\gamma > 0$ is used to balance the object proposal score and the symmetry score.

The proposals with P highest scores are selected for future recognition. Noted that we exclude the proposals which are too small to provide sufficient information for recognition. Only bounding boxes which satisfy the criteria that $b_w > \eta I_w$ and $b_h > \eta I_h$ will be considered, where $0 < \eta < 1$, b_w and b_h are the width and height of box b, I_w and I_h are the width and height of image I. Some returned top proposals by proposed method are shown in Fig. 4.5 (b).

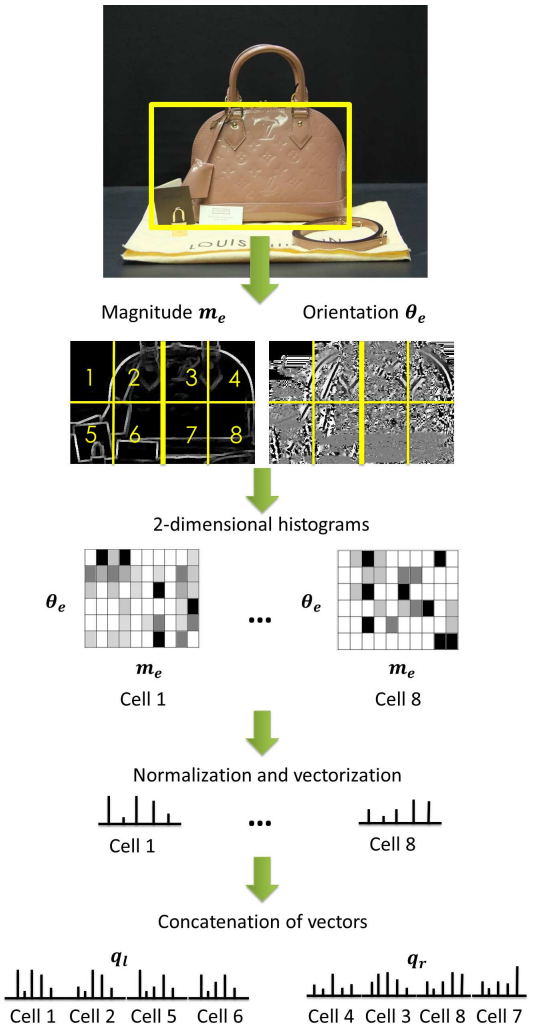


Figure 4.6: Feature extraction procedure for computing the symmetry score for a proposal. Each block in the 2-dimensional histogram indicates the frequency of occurrence of edge pixels in a cell with corresponding quantized magnitude and orientation, darker means higher frequency.

4.3.2 CNN Detection Model

In the CNN detection model, the deep CNN is trained as a binary classifier to distinguish the foreground handbag region from the background. Data preparation details will be discussed in the experiment.

4.3.3 Conditional Probability Model

Given a test image, its P top-ranked proposals are fed into both CNN detection model and FSCR-CNN classification model. For each of the top P proposals $r_i, i = 1, ..., P$, we compute the probability that r_i belongs to class j ($j = 1, ..., K$) as

$$c(j, f|r_i) = c(j|f, r_i)c(f|r_i), \tag{4.11}$$

where $c(j|f, r_i)$ indicates the classification score of the FSCR-CNN model for r_i with the assumption that it belongs to the foreground region, and $c(f|r_i)$ denotes the foreground detection score of the CNN detection model for r_i. The query image I is then predicted as class j^*, where

$$j^* = \arg \max_{\substack{j=1,...,K, \\ i=1,...,P}} (c(j, f|r_i)). \tag{4.12}$$

4.4 Experiments and Discussions

4.4.1 Experimental Setup

4.4.1.1 Data Preparation

We evaluated the proposed method on our branded handbag dataset introduced in Chapter 3. For the dataset, we use the cropped images as the input to train the framework, which is guided by the ground truth bounding box annotations.

For training the FSCR-CNN classification model, we augment the input data

by randomly sampling crops around the bounding box regions from both original images and their flipped versions. The cropping is done such that $T > 0.8$, where T is defined as the ratio between the intersection and the union of the crop and the bounding box. Eventually we generate $30,692$ images for training 401 handbag models in BrandBag, $17,774$ images for training 220 handbag models in BrandBag-I and $12,918$ images for training 181 handbag models in BrandBag-II.

For the CNN detection model, we regard the previously generated crops as positive data for training. To create the negative data, we randomly crop image patches from the background ($T < 0.6$) of the training images. Thus, we obtain $30,692$ positive and $34,472$ negative training data.

4.4.1.2 CNN Model

For all the CNN models we train, we start the training with a fixed learning rate and decrease it by a factor of 10 after the training error stops reducing. In our implementation, we use the MatConvNet toolbox [142], which provides different CNNs for computer vision applications. The CNN model proposed in [1] is incorporated in our proposed methods, unless otherwise specified.

4.4.2 Evaluation of the FSCR-CNN Classification Model

In this section, we evaluate the performance of the FSCR-CNN classification model for handbag recognition. In order to compare with the style-to-color discriminative representation (SCDR) framework proposed in Chapter 3, we use the ground truth bounding box annotation during testing. We have made two contributions in this model, which are FS-CNN and CR-CNN. The default values for parameters involved in the FSCR-CNN model are $\beta = 0.5$ and $\lambda = 0.01$, according to Section 4.4.3.

Table 4.1: Comparisons (Accuracy (%)) of handbag recognition frameworks SCDR (proposed in Chapter 3 vs. FSCR-CNN (proposed in this chapter) on the handbag datasets, given the ground truth bounding box annotations.

Method	Datasets		
	BrandBag	BrandBag-I	BrandBag-II
SCDR	84.01	92.77	72.25
CNN	71.27	84.01	60.69
FS-CNN	73.20	87.07	64.65
CR-CNN	77.52	90.54	66.72
FSCR-CNN	79.54	92.87	68.24
CNN-G	80.10	89.50	66.67
FS-CNN-G	83.51	92.82	70.41
CR-CNN-G	86.69	93.98	74.89
FSCR-CNN-G	87.97	95.60	76.44

As shown in Table 4.1, for the three handbag datasets, FS-CNN, CR-CNN and FSCR-CNN outperform CNN with around 3%, 6% and 8% improvement in accuracy, respectively. Compared with SCDR, FSCR-CNN performs slightly better on BrandBag-I, while worse on BrandBag and BrandBag-II. In order to update FSCR-CNN on recent networks, we also adopt the GoogLeNet [43] (the latest released pre-trained model). For GoogLeNet, we denote the original network as CNN-G, and the proposed corresponding networks are FS-CNN-G, CR-CNN-G and FSCR-CNN-G, respectively. It can be observed that SCDR outperforms CNN-G for all the datasets. While FSCR-CNN-G achieves better performance than the SCDR. Based on our observation, the intra-class variance of handbags in the BrandBag-I dataset is not as large as those in the BrandBag-II dataset. Due to the large variance and insufficient training data (data augmented based on 5 images/class for CNN models and 5 images/class for SCDR models), the

Table 4.2: Comparisons of handbag recognition frameworks SCDR vs. FSCR-CNN).

	SCDR (Chapter 3)	FSCR-CNN	FSCR-CNN-G (Chapter 4)
Hardware equipment	Desktop, CPU	Workstation, GPU	
Size of training data	5 images/model	data augmentation, 77 images/model	
Time cost for training	15 hours	19 hours	52 hours
Training Label Type	Handbag Model, Handbag SSC	Handbag Model	

performance of BrandBag-II is much lower than BrandBag-I.

In addition, we compare FSCR-CNN with SCDR from different aspects in Table 4.2. The two methods have pros and cons. Briefly speaking, SCDR needs less resources (hardware, training data, training time). However, it requires extra human labor, as it requires both the handbag model and the handbag SSC for the training images. FSCR-CNN requires GPU to speed up the training and testing procedure.

4.4.3 Evaluation of the Proposed Framework

In this section, the performance of the proposed handbag recognition framework is evaluated. Several parameters are needed to be set in advance. Similar to the parameter tuning in [143, 144], we tune our parameters with the help of cross-validation on the training data. Based on our observation, the top 10 proposals are sufficient to cover the handbag region. With this observation, we set the number of selected proposals $P = 10$ as the initial value. Percentage of non-discriminative feature elements β is within the range of $[0, 1]$. To disable feature selection during classifier training, we set $\beta = 1$ as the initial value. We first initialize γ, η and λ to be zero, then sequentially learn one after another by applying cross validation (e.g.,

Table 4.3: Comparisons of handbag recognition accuracies (%) on the handbag datasets.

Method	Datasets		
	BrandBag	BrandBag-I	BrandBag-II
EdgeBox + CNN classification (baseline)	51.54	65.63	42.91
Symmetry-based EdgeBox + CNN classification	59.57	73.83	47.99
Symmetry-based EdgeBox + CNN detection + CNN classification	68.01	81.96	52.77
Symmetry-based EdgeBox + CNN detection + FS-CNN classification	72.66	85.02	55.12
Symmetry-based EdgeBox + CNN detection + CR-CNN classification	73.92	89.32	58.39
Symmetry-based EdgeBox + CNN detection + FSCR-CNN classification	**75.18**	**90.18**	**60.10**

search for the best value of γ first, and then fix γ, and search for the best value of P). Unless otherwise specified, we use the default setting for these parameters.

In our framework, we use the foreground images to do the training. Thus, similar to the work in [48], we regard the object proposals + CNN classification as our baseline method, but we adopt a more recent edge box method [3] for extracting the object proposal. For each image, we extract the top P proposals and take the classification result of the object proposal which gives the maximum classification response. The performance of the baseline is given in Table 4.3, which is over 2% higher compared with using CNN only (49.22% for BrandBag, 59.48% for BrandBag-I and 40.82% for BrandBag-II).

4.4.3.1 Performance of the Symmetry-Based Proposal Localization

We evaluate our symmetry-based edge box method, which provides better proposals compared with the baseline. It is shown from the third row in Table 4.3 that we are able to obtain an improvement of over 5% in accuracy compared with the baseline for the three datasets.

4.4.3.2 Performance of the CNN Detection Model and Conditional Probability Model

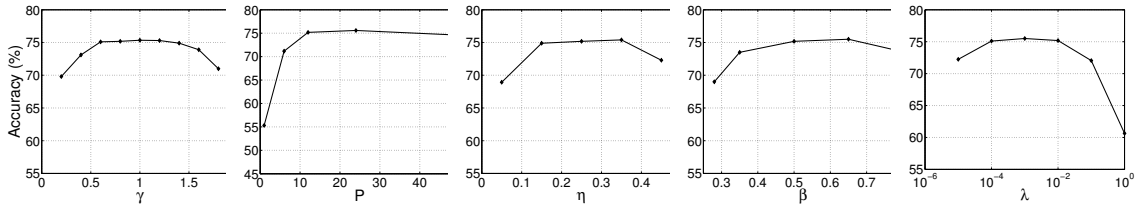
We recognize handbags with the CNN classification scores of their proposals conditioned on the CNN detection scores. The result of such handbag recognition is shown as Symmetry-based EdgeBox + CNN detection + CNN classification in Table 4.3. The combination of detection scores with classification scores by the conditional probability model provides better results with around 5% improvement over Symmetry-based EdgeBox + CNN classification.

4.4.3.3 Performance of the Proposed FSCR-CNN Classification Model

In Section 4.4.2, we summarized the performance of CNN, FS-CNN, CR-CNN and FSCR-CNN on handbag recognition, given the ground truth bounding box annotations during testing. In this part, we first report the handbag recognition accuracies after replacing existing CNN classification with our proposed classification models (FS-CNN and CR-CNN) separately in Table 4.3. Replacing CNN classification with FS-CNN classification and CR-CNN classification in our framework both leads to better results, with around 4% and 6% improvement in the accuracy respectively. We further combine FS-CNN and CR-CNN (i.e., FSCR-CNN) for the classification. With FSCR-CNN, our framework achieves over 20% better than the baseline (EdgeBox + CNN classification).

Table 4.4: Comparisons of handbag recognition accuracies (%) on the whole image for BrandBag, BrandBag-I and BrandBag-II.

Method	BrandBag	BrandBag-I	BrandBag-II
CNN	49.22	59.48	40.82
FS-CNN	50.98	62.66	42.27
CR-CNN	51.06	69.10	44.76
FSCR-CNN	52.07	71.98	46.56



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

Figure 4.7: Handbag recognition accuracies of Symmetry-based EdgeBox + CNN detection + FSCR-CNN classification when using different parameters: (a) tradeoff between the object proposal score and the symmetry score γ, (b) number of selected proposals P, (c) scale ratio for the selected proposals η, (d) percentage of non-discriminative feature elements β and (e) tradeoff between classification loss and regression loss λ.

We also evaluate the performance of applying only CNN and FSCR-CNN directly for handbag recognition (without foreground detection) in Table 4.4. It shows that FS-CNN and CR-CNN help to boost the performance and FSCR-CNN achieves the best accuracy for all the datasets. Therefore, it is beneficial to embed CNN models into our proposed framework for handbag recognition.

We vary the parameters γ, P, η, β and λ in the following ranges while keeping the others as the default values.

- $\gamma \in \{0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8\}$;
- $P \in \{1, 6, 12, 24, 48\}$;
- $\eta \in \{0.05, 0.15, 0.25, 0.35, 0.45\}$;

Table 4.5: Training time (in hour) of handbag dataset and testing time (in second) per handbag image of different methods.

Method	Training time (h)	Testing time (s/image)
EdgeBox	–	0.0465
Symmetry-based EdgeBox	–	0.0565
CNN detection	6.2384	0.0066
CNN classification	1.4691	0.0080
FS-CNN classification	11.9037	0.0075
CR-CNN classification	2.0394	0.0072
FSCR-CNN classification	19.4584	0.0074

- $\beta \in \{0.28, 0.35, 0.5, 0.65, 0.8\}$;
- $\lambda \in \{0.001, 0.01, 0.1, 1, 10, 100\}$.

We evaluate the sensitivity of each parameter for our proposed framework (i.e., Symmetry-based EdgeBox + CNN detection + FSCR-CNN classification) in Fig. 4.7. We observe that the performance is not sensitive to these parameters within certain ranges. λ is the most significant parameter in our method, as it balances the classification loss and regression loss. Accordingly, when λ is large (i.e., $\lambda > 1$), the soft label plays a more important role than the hard label for classification. When λ is small (i.e., $\lambda < 10^{-5}$), the effect of regression loss can be almost neglected.

4.4.3.4 Computational Complexity

Our framework consists of three parts: Symmetry-based EdgeBox, CNN detection and FSCR-CNN classification, where Symmetry-based EdgeBox has only testing phase, while the other two have both training and testing phases. We report the training time for 401 handbag classes (in hour) and testing time per image (in second) in Table 4.5. The experiment is conducted on MATLAB R2013a, in a

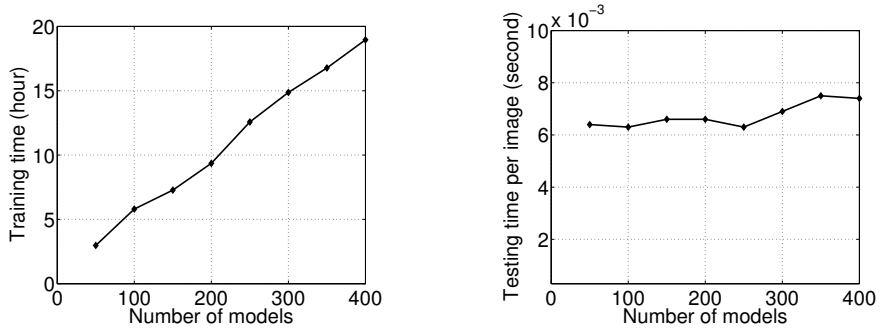
workstation of E5-2630 CPU, 96GB RAM, and a GPU Tesla K40. Note that for all CNN methods, we exclude the image loading time and set the batch size to 200 for each epoch. We also evaluate the complexity of FS-CNN and CR-CNN separately, and compare them with CNN. We observe that CR-CNN takes longer training time because it requires several more epochs to converge (25 to 30 for training handbags) than CNN (normally around 25 epochs). FS-CNN converges around 25 epochs. Besides, it takes longer time than CNN or CR-CNN for each epoch since the random forest implementation is time consuming. The training time of FSCR-CNN is longer, because it converges around 35 to 40 epochs. However, the training process can be applied off-line. To speed up, feature selection by random forest can be parallelized. In addition, with more GPUs, those CNNs can be also trained in parallel. For testing, the time costs of all CNNs are about the same.

In addition, we report the scalability of FSCR-CNN in terms of computational complexity. Fig. 4.8(a) shows the increase in the total training time vs. the increase in the number of handbag models. Fig. 4.8(b) plots the testing time for each image. We observe that with the increase in the number of training classes, the training time is somewhat linear. Noted that this is due to the increasing number of batches required at each epoch. The class number does not heavily influence the testing time per image.

4.4.4 Evaluation of the Generality of the Proposed FSCR-CNN Model

To verify the generality and superiority of our proposed FS-CNN and CR-CNN model over the CNN model, we also apply them on other fine-grained datasets: Oxford Flowers [5], Stanford Dogs [6] and Caltech-UCSD Birds [7].

For the Oxford Flowers dataset, we follow the data augmentation method pro-



(a) (b)

Figure 4.8: Training and testing time based on different number of handbag models for FSCR-CNN: (a) overall training time (hour), and (b) testing time for each image (second).

Table 4.6: Top-1 and top-5 accuracy (%) of CNN based architectures on the Oxford Flowers dataset [5], the Stanford Dogs dataset [6] and the UCSD-Birds dataset [7].

Method	Oxford Flowers		Stanford Dogs		UCSD-Birds	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
CNN	77.04	91.95	52.41	81.12	59.94	83.34
FS-CNN	79.69	93.71	51.40	80.73	64.58	87.04
CR-CNN	81.83	93.53	**64.79**	**88.73**	67.38	88.07
FSCR-CNN	**82.21**	**94.29**	–	–	**69.78**	**89.54**

vided by [129]. 16 representatives for each image without segmentation (original image, 5 crops, 2 rotation and their mirrors) are built. The top-1 and top-5 accuracies reported for CNN, FS-CNN, CR-CNN and FSCR-CNN are listed in the 2^{nd} and 3^{rd} column of Table 4.6. The existence of background with green grasses affects the color of flowers. Nevertheless, color component is still of a certain importance for classifying flowers. FS-CNN and CR-CNN both perform better than CNN, and the FSCR-CNN achieves the best.

For the Stanford Dogs dataset, we apply the bounding box annotations for both training and testing procedure as indicated in [128]. For all CNN networks, we randomly crop around the bounding box regions and keep the crops with $T > 0.8$.

Table 4.7: Comparisons with other leading fine-grained object recognition approaches on the Stanford Dogs dataset [6].

Method	Accuracy (%)	Mean accuracy (%)
Yang et al. [128]	38.00	–
Pu et al. [145]	39.30	–
Chai et al. [146]	–	45.60
Gavves et al. [65]	–	50.10
CNN	52.41	51.08
CR-CNN	**64.79**	**63.23**

Eventually, the data augmentation is done by making an average of 9 representations for each training image. The fourth and fifth column of Table 4.6 shows the comparisons of the top-1 and top-5 accuracies for different CNN models. The proposed CR-CNN outperforms CNN with at least 12% improvement, which even surpasses the previously published results (see Table 4.7). However, FS-CNN is not helpful in improving the performance. This is due to the reason that the color is not a sensitive feature for dogs. In this dataset, some dogs even wear clothes or heavily occluded. Therefore, in the following experiments for dogs, we will not evaluate the FS-CNN and FSCR-CNN models unless otherwise specified.

Like the Oxford Flowers dataset, we follow the data augmentation method [129] for the Caltech-UCSD Birds-200-2011 dataset. Results are shown in the last two columns of Table 4.6. Like flowers, color is an important component for classifying birds. Therefore, FS-CNN is also helpful for bird recognition. Compared with CNN, FSCR-CNN improves the recognition accuracy significantly by 10% for Top-1 accuracy.

Nowadays different CNN architectures have been designed [43, 45, 47]. We also embed our proposed Feature Selection or joint Classification-Regression on a well performed architecture CNN-S [47]. Similarly, the corresponding networks are

Table 4.8: Top-1 and top-5 accuracy (%) of CNN-S based architectures on the Oxford Flowers dataset [5], the Stanford Dogs dataset [6] and the UCSD-Birds dataset [7].

Method	Oxford Flowers		Stanford Dogs		UCSD-Birds	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
CNN-S	81.09	94.02	71.27	93.62	65.79	88.25
FS-CNN-S	82.05	94.49	–	–	67.10	88.82
CR-CNN-S	85.15	95.10	**76.64**	**94.92**	71.73	89.63
FSCR-CNN-S	**86.21**	**95.38**	–	–	**73.90**	**91.23**

denoted as FS-CNN-S, CR-CNN-S and FSCR-CNN-S. CNN-S is similar to the OverFeat structure [44]. Similar to Section 4.4.2, we also update our proposed FSCR-CNN on GoogLeNet [43].

Table 4.8 shows the performance comparisons among FS-CNN-S, CR-CNN-S, FSCR-CNN-S and CNN-S on the Oxford Flowers dataset and the UCSD-Birds dataset, as well as the comparison between CR-CNN-S and CNN-S on the Stanford Dogs dataset. Again, the CR-CNN-S model performs better on these fine-grained object datasets, with over 4% increase in accuracy compared with CNN-S. FS-CNN-S and FSCR-CNN-S are able to achieve better performances on the flower and bird dataset. Compared with CNN-S, FSCR-CNN-S achieves over 5% improvement in accuracy. The comparisons of CNN-G with FSCR-CNN-G for the three datasets are summarized in Table 4.9. We observe that the improvements in classification accuracy are not limited to CNN structures, as the accuracies are further boosted by over 5%.

CNN-S and CNN-G could also be incorporated into our proposed framework (to replace CNN) for handbag recognition. Table 4.10 compares our proposed framework (i.e., Symmetry-based EdgeBox + CNN-S (or CNN-G) detection + FSCR-CNN-S (or FSCR-CNN-G) classification) with the baseline structure (i.e.,

Table 4.9: Comparisons of CNN-G and FSCR-CNN-G on the Oxford Flowers dataset [5], the Stanford Dogs dataset [6] and the UCSD-Birds dataset [7].

Method	Oxford Flowers		Stanford Dogs		UCSD-Birds	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
CNN-G	85.46	95.77	73.91	94.26	75.73	92.66
FSCR-CNN-G	**90.43**	**97.06**	**78.91**	**94.98**	**81.74**	**94.93**

Table 4.10: Comparisons of handbag recognition accuracies (%) on different frameworks of CNN-S and CNN-G-based architecture.

Framework	Datasets		
	BrandBag	BrandBag-I	BrandBag-II
EdgeBox + CNN-S classification	62.47	78.92	41.61
Symmetry-based EdgeBox + CNN-S detection + FSCR-CNN-S classification	73.65	92.79	55.64
EdgeBox + CNN-G classification	65.78	83.40	45.07
Symmetry-based EdgeBox + CNN-G detection + FSCR-CNN-G classification	**80.33**	**94.48**	**66.69**

EdgeBox + CNN-S (or CNN-G) classification). Since the intra-class variance of handbags in BrandBag-II dataset is much larger than that in BrandBag-I due to illumination changes, distortion and varying viewpoint rotations. With limited training data, the recognition accuracy of BrandBag-II is much lower than that of BrandBag-I. It can be seen that our proposed framework can boost the handbag recognition performance by more than 10%.

4.5 Summary

In this chapter, a novel CNN model, named FSCR-CNN, is proposed for handbag recognition. FSCR-CNN model attenuates the illumination changes and inter-class similarity among handbags. We also design an end-to-end handbag recognition

framework. In this framework, we first incorporate the symmetry property of the handbag for extracting handbag proposals. Then each proposal is fed into a CNN detection model and a proposed FSCR-CNN classification model. Proposal detection scores and classification scores are eventually combined by a conditional probability model to further improve the performance of handbag recognition. Extensive experiments show that FS-CNN is helpful at recognizing color sensitive fine-grained objects (3% improvement on the handbag dataset, 2% improvement on the Oxford Flowers dataset and 5% improvement on the UCSD-Birds dataset) and CR-CNN performs fairly well on fine-grained object recognition tasks, with 8% improvement for the handbag dataset, 4% improvement for the Oxford Flowers dataset, 12% improvement for the Stanford Dogs dataset and 7% improvement for the UCSD-Birds dataset. The experimental results on our newly constructed handbag datasets verify the advantages of each component of our framework.

Chapter 5

Handbag Recommendation

This chapter considers to recommend handbags to a shopper, based on the handbag images the shopper has clicked, which can also be embedded into a practical multimedia system for online advertising and commerce[1]. Recommendation is performed by Joint learning of attribute Projection and One-class SVM classification (JPO). More specifically, for the handbag images clicked by each shopper, we project the original image feature space onto a compact attribute space. The projection matrix is learned jointly with a one-class SVM to yield a shopper-specific one-class classifier. Results show that the proposed JPO handbag recommendation performs favorably based on the initial subject testing. We also propose a post-processing: a weighted AutoEncoder technique which refines the recommended results by detecting the handbags that deviate from others. This post-processing is regarded as an extension of handbag recommendation which can be applied for other outlier detection problems as well.

5.1 Introduction

Purchasing handbags online is convenient and efficient. However, for some e-commerce websites or manufacturers, how to find the shopper's most preferred

[1]Part of this chapter was presented in [147]

handbags is an interesting while challenging recommendation problem. The challenges lie in the sheer variety of different styles, as well as customers' tastes and fashion elements.

In this chapter, we aim to address the image-based handbag recommendation problem by analyzing the content of the preferred handbags, i.e., the feature of handbag image itself rather than the ratings from other shoppers. Our focus is to capture the common property of those handbags. Empirically, shoppers tend to prefer handbags with similar (or even the same) attributes, such as the shape, pattern, color or style for a certain period of time. Defining attributes for handbags is somewhat difficult. Besides, it takes tedious human labor to label the attributes.

To address these issues, we first extract general features (e.g., SIFT [31], LBP [32]) from the handbag images. Then we project the features onto an attribute space automatically by an attribute projection model. Our work denotes "attribute" as shareable properties of handbags which may not have concise semantic names [148]. Moreover, instead of directly feeding the extracted attribute features into a shopper-specific classifier to make a recommendation, we predict whether the handbag is preferred by a shopper via a joint learning of the projection model and one-class classification model of the attributes.

5.2 Handbag Recommendation

In a recommender system, the problem of predicting whether a product is preferred by the shopper can be modeled as a binary classification task. Shoppers' behaviors such as click or bookmark offer certain positive data [149]. However, if a product is not clicked by the shopper, it can be interpreted into two ways. One way is that the shopper is not interested in the product. The other is that the shopper does

not know about the product. In this chapter, we consider the positive data, and formulate the recommendation as a one-class classification problem.

One-class SVM [150, 151] is a popular one-class classifier. However, original image features may not capture the shareable properties of a shopper's preference due to the diversity of handbags. It is not appropriate to feed original features into a one-class SVM directly. For online shopping, the most direct way to recommend handbags is to return the ones with the same (or similar) attributes as those chosen by the shopper. Feed the attributes extracted from positive images into one-class SVM is direct and simple.

We believe that there exists some common attributes among the preferred handbags clicked by each shopper. Therefore, we propose to project features from the original feature space into an attribute space for extracting the common properties of positive data. We jointly learn the projection and one-class SVM classification to build the classifier for handbag recommendation. Next, we briefly review the one-class SVM, followed by the introduction of our joint learning algorithm.

5.2.1 One-Class SVM

One-class SVM samples from positive class and aims at finding minimal circumscribing hyperball in a high-dimensional space. If the newly encountered data is too far from the learned hypeball, it will be labeled as out-of-class, otherwise it is regarded as in-class. This approach is similar with the density estimation [152] because it captures regions where the probability density of the data stays. Given a set of positive data $\{\boldsymbol{x}_i\}_{i=1}^N$, the quadratic program minimization function [150] of linear one-class SVM is:

$$\min_{\boldsymbol{\omega},\xi_i,\rho} \frac{1}{2}||\boldsymbol{\omega}||^2 + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho, \tag{5.1}$$

subject to:

$$\langle \boldsymbol{\omega}, \boldsymbol{x_i} \rangle \geq \rho - \xi_i,\ \xi_i \geq 0.$$

where $\boldsymbol{\omega}$ is a weight vector, ρ is an offset parameterizing the hyperball, ν decides the tradeoff between maximizing the margin and slack variables ξ_i allow some data points to lie within the margin. $<,>$ denotes the inner product of the vectors.

5.2.2 Joint Learning of Attribute Projection and One-Class SVM Classification

In this section, we aim to learn a projection that can automatically project the original features onto an attribute space, and meanwhile learn a hyperball to separate the learned attribute features from the origin (negative class). The learned attribute features may not necessarily be semantically meaningful, but they represent some projections which capture the common properties of all features in the positive class. Let $\{\boldsymbol{x}_i\}_{i=1}^N$ be the set of original features extracted from the shopper-clicked images, $\boldsymbol{x}_i \in R^{N\times 1}$ and $\boldsymbol{\omega} \in R^{M\times 1}$, the objective function is defined as

$$C = \frac{1}{2N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} ||\boldsymbol{A}\boldsymbol{x}_i - \boldsymbol{A}\boldsymbol{x}_j||^2 +$$

$$\lambda \left(\frac{1}{2}||\boldsymbol{\omega}||^2 + \frac{1}{\nu N} \sum_{i=1}^{N} (\rho - \boldsymbol{\omega}^T \cdot \boldsymbol{A} \cdot \boldsymbol{x}_i)^2 - \rho \right) + \frac{\gamma}{2}||\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{I}||^2. \tag{5.2}$$

It has three terms, where the first term is the attribute-based projection term. $\boldsymbol{A} \in R^{M\times H}$ is a learned projection matrix, which projects H-dimensional original features onto M-dimensional attribute features, where $M \ll H$. The second term is the attribute-based one-class SVM classification. In this term, λ is a scalar that trades off between this term with others. $(\boldsymbol{\omega}, \rho)$ are weight vector and learned hyperball for classification based on attribute features respectively. ν is a smooth-

ness parameter, and it trades off model losses on the positive data with other components. This term separates the positive data from the origin. We borrow the idea from one-class SVM, but use square loss rather than hinge loss as in (5.1). The least square model has an exact closed form solution instead of a quadratic programming and it learns the hyperball closely to target values. The component $-\rho$ is to ensure a hyperball far from the origin, which sets an upper bound on the fraction of outliers. And the last term is the regularization term, which is used to force the projection matrix to be orthogonal. γ is a trade off parameter.

JPO minimizes two types of errors, one is feature-to-attribute error and the other is attribute-to-preference error. We jointly learn the projection matrix and classifier to make the distances among learned attribute features close, such that the common properties of positive data are captured. On the other hand, the classifier leads to a more proper projection matrix. Meanwhile, the learned projection matrix in turn helps the one-class classifier learning.

5.2.3 Optimization

Our objective function is not convex. However, when $\boldsymbol{A}$ is fixed, the objective function is convex regarding $(\boldsymbol{\omega}, \rho)$, and when $(\boldsymbol{\omega}, \rho)$ is fixed, a suboptimal $\boldsymbol{A}$ can be obtained by a gradient descend technique. Hence we employ an alternating optimization algorithm to iteratively update $\boldsymbol{A}$ and $(\boldsymbol{\omega}, \rho)$.

First, we consider the $(\boldsymbol{\omega}, \rho)$-subproblem of (5.2) with $\boldsymbol{A}$ fixed. This problem has an optimum closed form solution by taking the derivative with respect to $\boldsymbol{\omega}$ and ρ:

$$\frac{\partial C}{\partial \boldsymbol{\omega}} = \lambda \boldsymbol{\omega} + \frac{2\lambda}{\nu N} \left(\boldsymbol{\omega} \boldsymbol{A} \boldsymbol{X} \boldsymbol{I}^T \boldsymbol{X}^T \boldsymbol{A}^T - \rho \boldsymbol{d} \boldsymbol{X}^T \boldsymbol{A}^T \right), \tag{5.3}$$

and

$$\frac{\partial C}{\partial \rho} = \frac{\lambda}{\nu N}(2N\rho - 2\boldsymbol{\omega}\boldsymbol{A}\boldsymbol{X}\boldsymbol{d}^T) - \lambda. \tag{5.4}$$

where $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N) \in R^{H\times N}$ is a matrix composed of positive training data. $\boldsymbol{I} \in R^{N\times N}$ is an identity matrix. $\boldsymbol{d} = (1, ..., 1) \in R^{1\times N}$ is a row vector.

Second, we fix $(\boldsymbol{\omega}, \rho)$ to update $\boldsymbol{A}$, the cost can be re-written in the matrix form as:

$$C = \frac{1}{N-1}\left(Tr(\boldsymbol{X}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{X}) - \frac{1}{N}\boldsymbol{d}\boldsymbol{X}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{X}\boldsymbol{d}^T\right)$$
$$+\frac{\lambda}{\nu N}\left(N\times\rho^2 + \boldsymbol{\omega}\boldsymbol{A}\boldsymbol{X}\boldsymbol{I}\boldsymbol{X}^T\boldsymbol{A}^T\boldsymbol{\omega}^T - 2\rho\boldsymbol{\omega}\boldsymbol{A}\boldsymbol{X}\boldsymbol{d}^T\right) - \lambda\rho + \frac{\gamma}{2}||\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{I}||^2. \tag{5.5}$$

where $Tr(\cdot)$ denotes the trace of the matrix, and the gradient is written as:

$$\frac{\partial C}{\partial \boldsymbol{A}} = \frac{1}{N(N-1)}\boldsymbol{A}\boldsymbol{X}\boldsymbol{G}\boldsymbol{X}^T + \frac{\lambda}{\nu N}2\boldsymbol{\omega}^T\boldsymbol{\omega}\boldsymbol{A}\boldsymbol{X}\boldsymbol{I}\boldsymbol{X}^T -$$
$$\frac{2\rho\lambda}{\nu N}\boldsymbol{w}^T\boldsymbol{d}\boldsymbol{X}^T + 2\gamma\boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{I}\right). \tag{5.6}$$

where $\boldsymbol{G} = 2N\boldsymbol{I} - 2\boldsymbol{Q}$ and $\boldsymbol{Q} \in R^{N\times N}$ is an all one matrix.

Here, we adopt L-BFGS [153] to optimize our cost function. It is a limited-memory quasi-Newton algorithm for unconstrained optimization. The elements of the projection matrix $\boldsymbol{A}$ is first initialized with all zeros, except that for those corresponding to the feature dimension with small variance, we set it to one. More specifically, we rewrite the training data matrix according to different feature dimensions: $\boldsymbol{X} = \left(\boldsymbol{z}_1^T, ..., \boldsymbol{z}_H^T\right)^T$. Then we rank $\{\boldsymbol{z}_j\}_{j=1}^H$ according to the ascending order of their variances. We denote the ranking as $R = (r(1), ..., r(H))$ where $r(j) \in \{1, ..., H\}$ indicates the rank of the variance of feature $\boldsymbol{z}_{r(j)}$ is j. After

Algorithm 1 JPO: Joint learning of attribute projection model and one-class classification model

Input:
$\boldsymbol{X}$: Positive training data
λ, γ: Trade off parameters for controlling weight of attribute-based projection term, attribute-based one-class SVM classification term and regularization term
ν: Trade off parameters for controlling losses on the positive data
Output:
$\boldsymbol{A}$: Projection matrix
$(\boldsymbol{\omega}, \rho)$: weight vector and parameter for characterizing the hyperball
Initialize $\boldsymbol{A}$ based on (5.7)
while $\boldsymbol{A}$ and $(\boldsymbol{\omega}, \rho)$ not converge **do**
    Fix $\boldsymbol{A}$ and solve (5.3)-(5.4) by updating $(\boldsymbol{\omega}, \rho)$
    Fix $(\boldsymbol{\omega}, \rho)$ and solve (5.6) by updating $\boldsymbol{A}$ via gradient descend: $\boldsymbol{A} \leftarrow \boldsymbol{A} - \beta \frac{\partial C}{\partial \boldsymbol{A}}$
end while
return $\boldsymbol{A}$, $(\boldsymbol{\omega}, \rho)$

initialization, the elements in $\boldsymbol{A}$ are zero except that:

$$\boldsymbol{A}(i, r(i)) = 1 \quad \text{for } i = 1, ..., M. \tag{5.7}$$

At each iteration, we alternatively update $(\boldsymbol{\omega}, \rho)$ and $\boldsymbol{A}$ until convergence. Experimentally, it shows that the algorithm converges within dozens of iterations. The learning algorithm procedure is shown in Algorithm 1.

For any testing image, we first project its original feature onto the attribute feature, then the learned attribute-based one-class SVM is applied to obtain the prediction score.

5.3 Post-Processing: Weighted AutoEncoder Outlier Detection

In this section, an outlier detection technique is applied to detect the handbags which are significantly different from the others in the recommendation list. Those

outliers can be removed if required.

An outlier (anomaly) is an observation which deviates from the other observations [154]. A large amount of corpus has proposed various techniques for outlier detection or removal. These methodologies include density estimation [155–157], hyperplane learning [150, 158, 159] and data reconstruction. The density-based (or distance-based) outlier detection considers the ratio of the density around an object and the density around its neighbors. Methods such as local outlier factor (LOF) [155] compares the density estimate for each object inside a dataset with the average density estimate for the object's nearest neighbors. Kernel density estimator (KDE) and its derivatives [156, 157] are non-parametric methods to estimate the probability density function of a continuous random variable.

Hyperplane-based methods, such as one-class SVM [150, 158], sample from positive class and aim at finding minimal circumscribing hyperplane. It can tolerate a small quantity of outliers by introducing the slack variable, which allows an uncertain input data to lie on the other side of the decision boundary. Unsupervised one-class learning (UOCL) [159] proposes a joint learning method. It learns a soft label assignment for outliers and inliers and a one-class classifier.

Reconstruction-based methods compute the anomaly score by comparing the reconstruction residues. Methods such as PCA, Kernel PCA, Robust PCA [160, 161], sparse coding and popular replicator neural network (AutoEncoder) reproduce the input and identify samples taking on high reconstruction residues as outliers. The shortcoming of the aforementioned methods is that most of them need to follow the rule about "few and different" for detecting outliers. When the data set is corrupted by a large fraction of outliers, they may not work well.

Motivated by the AutoEncoder [2], we propose a Weighted AutoEncoder to train the network by reducing the influence of dubious outliers iteratively. This

works well even when the data set consists of a large fraction of outliers (e.g., 50%).

5.3.1 AutoEncoder

An unsupervised AutoEncoder neural network [2] is composed of multiple encoder layers and decoder layers, which encourages sparsity and enforces the outputs to be equal to the inputs. The feedforward procedure transforms the high-dimensional data into a low-dimensional code in an encoder network and recovers the data from the code in a decoder network.

The encoder layer and decoder layer have the same mathematics form:

$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} \\ \boldsymbol{a} &= s(\boldsymbol{z}) \end{aligned} \tag{5.8}$$

where $\boldsymbol{x} \in R^{d\times 1}$ is an input vector, $\boldsymbol{a} \in R^{r\times 1}$ is the output vector and s is the activation function, such as the sigmoid function. $\boldsymbol{W} \in R^{r\times d}$ and $\boldsymbol{b} \in R^{r\times 1}$ are network parameters, where the former one is a weight matrix, and the latter one is a bias vector. For encoder layer, $\boldsymbol{x}$ is the original data and $\boldsymbol{a}$ is the encoded hidden data; In contrast, for decoder layer, $\boldsymbol{x}$ is the encoded hidden data and encoded hidden data is the reconstructed data. In an AutoEncoder neural network, the outputs of each layer is wired to the inputs of the successive layer, so we can write the forward propagation of the network in an iterative way. Formally, consider an AutoEncoder network with n layers and let $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$ be the parameters of the l-th layer, we have

$$\begin{aligned} \boldsymbol{z}^{(l+1)} &= \boldsymbol{W}^{(l)}\boldsymbol{a}^{(l)} + \boldsymbol{b}^{(l)} \\ \boldsymbol{a}^{(l+1)} &= s(\boldsymbol{z}^{(l+1)}) \end{aligned} \tag{5.9}$$

The optimization objective that penalizes the reconstruction error between the

input of the network $\boldsymbol{a}^{(l)} = \boldsymbol{x}^{(1)} = (x_k^{(1)}; k = 1, \ldots, d)^T$ and $\boldsymbol{a}^{(n)} = (a_k^{(n)}; k = 1, \ldots, d)^T$ can be measured as a cross-entropy of the reconstruction:

$$L(\boldsymbol{x}^{(1)}, \boldsymbol{a}^{(n)}) = -\sum_k^d \Big(x_k^{(1)} \log a_k^{(n)} + (1 - x_k^{(1)}) \log(1 - a_k^{(n)}) \Big)$$

5.3.2 Weighted AutoEncoder

As an AutoEncoder learns to minimize the reconstruction error of each data equally, it is not specially designed for removing outliers. A large fraction of outliers could result in impaired performance. We propose to discover dubious anomalies gradually by an automatic weighting procedure, such that the outliers will become less relevant for the optimization. We add such adaptive weights α on top of the cross-entropy objective function in Eq. (5.10)

$$\mathcal{L}(\boldsymbol{x}^{(1)}, \boldsymbol{a}^{(n)}) = -\sum_k^d \alpha \Big(x_k^{(1)} \log a_k^{(n)} + (1 - x_k^{(1)}) \log(1 - a_k^{(n)}) \Big), \tag{5.10}$$

where α is an adaptive weight, computed by the normalized reconstruction error $e \in [0, 1]$ of previous iteration for this data

$$\alpha = \frac{1}{1 + \exp\big(-\beta(\frac{1}{e} - \gamma)\big)}. \tag{5.11}$$

Here we adopt the sigmoidal membership function. In our implementation, parameter $\beta = 100$ is to generate a steep slope and polarize α to be near 0 or near 1. $\gamma = 0.5$ sets the coordinate point $(1, 0)$ as the center of symmetry.

The standard back-propagation can be applied to optimize the parameters of the weighted AutoEncoder network. First, for the output layer (layer n), set

$$\delta^{(n)} = \nabla_{\boldsymbol{a}^{(n)}} \mathcal{L} \cdot s'(\boldsymbol{z}^{(n)}), \tag{5.12}$$

where $\nabla_{\boldsymbol{a}^{(n)}} \mathcal{L} = (-\alpha \frac{x_k^{(1)} - a_k^{(n)}}{a_k^{(n)}(1 - a_k^{(n)})}; k = 1, \ldots, d)^T$ and $s'(\boldsymbol{z}^{(n)}) = s(\boldsymbol{z}^{(n)})(1 - s(\boldsymbol{z}^{(n)}))$

if s is a sigmoid function. Then, for $l = n-1, n-2, n-3, \ldots, 2$, set

$$\delta^{(l)} = \left((\boldsymbol{W}^{(l)})^T \delta^{(l+1)}\right) \cdot s'(\boldsymbol{z}^{(n)}). \tag{5.13}$$

At last, the partial derivatives of the cross-entropy function w.r.t. the parameters of the network can be obtained by

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}^{(l)}} &= \delta^{(l+1)} (\boldsymbol{a}^{(l)})^T \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{b}^{(l)}} &= \delta^{(l+1)} \end{aligned} \tag{5.14}$$

5.3.3 Outlier Detection

After applying the proposed weighted AutoEncoder, the reconstruction error, termed as anomaly score of each data is computed as:

$$\varepsilon = \| \boldsymbol{a}^{(n)} - \boldsymbol{x}^{(1)} \|_2^2 \tag{5.15}$$

We sample the images in this data set by sorting their anomaly scores obtained from the weighted AutoEncoder. In the experiment part, we will show that the recommended handbags can be re-ordered according to their anomaly scores. The recommended handbags with high anomaly scores are taken as the outliers and could be removed from the recommendation list if required. We also evaluate the proposed WAE in terms of outlier detection in one public image dataset UIUC-Scene [8].

5.4 Experiments and Analysis

5.4.1 Dataset Construction

To the best of our knowledge, no existing handbag dataset for recommendation purpose is available. In order to compare our JPO algorithm with other ap-

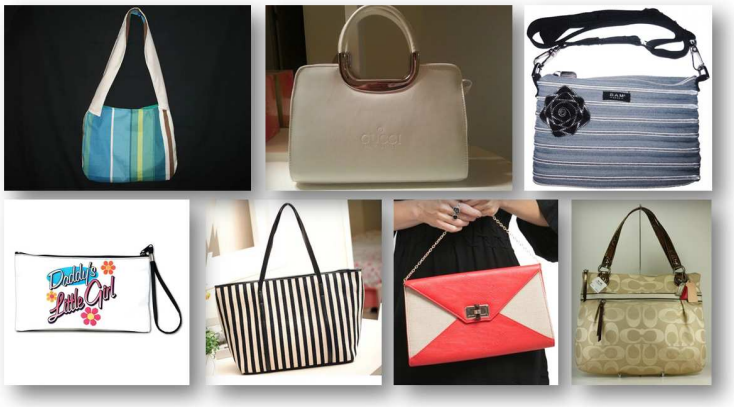


Figure 5.1: Examples of handbag images in our dataset.

proaches, we create a handbag dataset which consists of 835 handbag images from the shopping websites like *Amazon.com*. Some examples are given in Fig. 5.1. The handbags in our dataset have large variations due to diverse colors, patterns or styles.

As the preference of each shopper is different, we asked 30 subjects (with different nationalities, and age ranges from 19-35) to choose handbags that they like from the dataset. To imitate the online shopping scenario, we reshuffle the handbags for each subject and choose a display of 10 images to show each timestep. The subject can leave the system at any timestep. On one hand, a shopper wants to obtain some recommendation results only with a small number of clicks. On the other hand, we need sufficient images for training and testing, we retain the subjects' data containing at least 15 handbag images (10 for training and more than 5 images are used as positive data for testing).

5.4.2 Experimental Settings

We build a model per user, and among the handbag images clicked from each subject, we randomly sample 10 images as positive data for training. The rest

Figure 5.2: Sampled selected handbags from some subjects, where one row indicates handbags clicked by one subject.

of the dataset is for testing. Fig. 5.2 shows some clicked handbag images from different subjects. Observed that although the handbags selected by a subject contain some diversity, they roughly share some common properties. Such as the denim hobo bags, plaid pattern, plain pattern, horizontal pattern, and cartoon style.

In our experiment, instead of applying feature extraction methods from the whole handbag image, we first adopt a saliency detection technique [162], and then extract the bag-of-words [8] (1^{st} level pyramid only, codebook size 256) of SIFT feature [31] and color naming feature [115] from the bounding box covering top 10 saliency windows. This feature vector is regarded as the original feature and fed into all content-based comparison methods.

Among the set of 26 subjects' data, in order to set the parameters, 10 are randomly chosen to form a validation set. The remaining 16 are used as testing

data. We use the grid search to find that $\lambda = 4$, $\gamma = 0.1$ and $\nu = 0.1$ give good performance for JPO on the validation dataset. We also find that by setting $M \ll H$ (such as $M = 2$ compared with $H = 267$), we can obtain good results for the handbag recommendation.

5.4.3 Evaluation Protocol

We present each subject (one of the 16 subjects for testing) with K handbags in our dataset except for the 10 training handbags, sorted by the subject's predicted scores. As suggested in [80], we evaluate *recall@K* based on which of these handbags were selected by the subject. The definition of *recall@K* [80] is

$$\text{recall@}K = \frac{\text{number of handbags the subject likes in top } K}{\text{total number of handbags the subject likes}} \tag{5.16}$$

The recall values are averaged for all 16 subjects and plotted for each top K recommended handbags. For each subject, we list the subject's mean recall value for comparison. We also evaluate our algorithm by precision-recall curves as they are widely used in information retrieval and they give an informative picture of the algorithms [163].

5.4.4 Comparisons

We test and compare different recommender algorithms on the remaining 16 subjects' data, including the content-based methods: proposed JPO, one-class SVM (OC-SVM) and a public semi-supervised learning method [164] (WELLSVM-SSL). Semi-supervised learning is also a good choice for the recommender systems [165]. We compare with a conventional recommender relying solely on user-item matrix, which is the item-based collaborative filtering (CF) [166]. Recent researches sug-

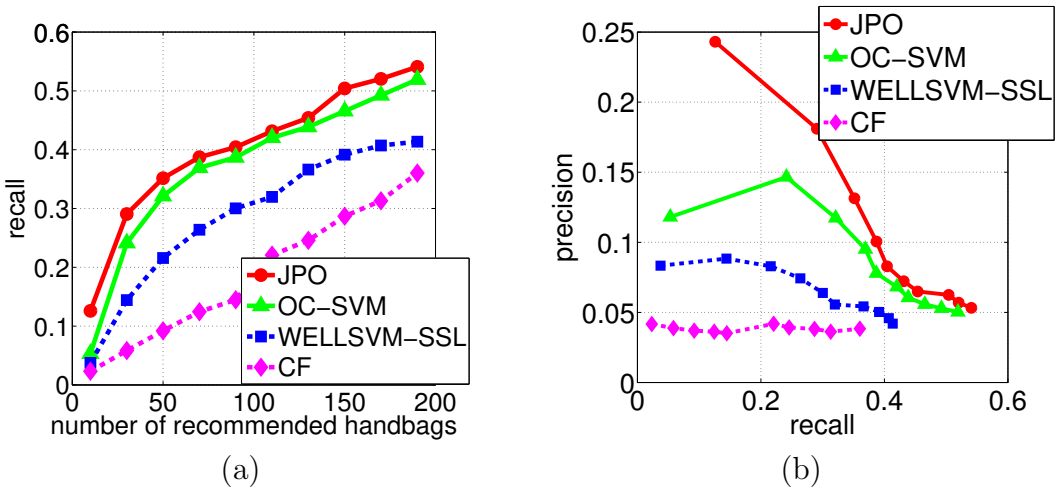


Figure 5.3: Comparisons of (a) recall curves and (b) precision-recall curves by varying the number of returned handbags over all 16 subjects for testing.

gest that the popular baseline is often difficult to beat. For OC-SVM, it inputs the original feature from 10 positive images and trains a one-class classifier. In WELLSVM-SSL, we train a semi-supervised model for each subject, where the labeled data is the same as OC-SVM and unlabeled data are original features from all the other $835 - 10 = 825$ images. We apply the cosine similarity and weighted sum to obtain the predictions in CF.

Fig. 5.3 (a) shows the recall curves for handbag recommendation over all 16 subjects, where we vary the number of recommended handbags $K = 10, 30, ..., 200$. It can be seen that our method performs the best for those subjects. We show some recommended handbags given by JPO and OC-SVM for subject #1 as an example (see Fig. 5.4). Compared with the handbags recommended by OC-SVM, those recommended by the proposed JPO are closer with the handbags selected by the subject (first row of Fig. 5.2). The reason is that OC-SVM may not well capture the common properties of the preferred handbags from merely the original features, as the distances among the original features are large. The proposed

(a)

(b)

Figure 5.4: Handbags recommended by (a) proposed JPO and (b) OC-SVM in the top ranks for subject #1.

JPO jointly learns the feature and the classifier, which minimizes the distances among newly learned features (potential attributes) while learning a more reliable classifier to differentiate the positive data with others. As expected, CF does not work well due to the small data size.

In the experiment, we observe that subject #1 has a high recall. The reason is that the subject clicks similar types of handbags (Stonewashed Denim Hobo bags), which result in a denser distribution of original features and easier-to-learn attributes than other subjects. For other subjects, however, the selected handbags have a certain degree of diversity. We also investigate the learned attribute features for positive training data (i.e., $\boldsymbol{AX}$ in Eq. (5.3)-(5.4)) of different subjects, and find that usually a higher value in $\boldsymbol{AX}$ yields a higher performance. A high $\boldsymbol{AX}$ indicates high consistency among the appearances of the chosen handbags.

We then vary the number of returned handbags $K = 10, 30, ..., 200$ and draw an average precision-recall curve for all 16 subjects. As shown in Fig. 5.3 (b), the proposed JPO can always outperform OC-SVM, WELLSVM-SSL and CF methods, especially when recall is small. The experiment shows an overall low

precision, which may be caused by the uncertainty of the unlabeled handbags.

5.4.5 Evaluation on Outlier Detection

In this part, we evaluate the performance of the weighted AutoEncoder on outlier detection. Suggested by [159, 167], instead of testing the number of outliers predicted correctly, we examine the likelihood of an image not being an outlier. We firstly deal with the handbag recommendation problem. Then the weighted AutoEncoder is further verified on the UIUC-Scene dataset [8], as suggested in [159]. In this experiment, we extract CNN features by using the pre-trained model [47] based on the ImageNet dataset.

Studies show that 3 out of 4 shoppers never scroll past the first page of recommended results and give up the trail if he/she cannot find the desired product. Therefore, for each subject, we only focus on the top results. We extract the top 40 recommended handbags and together with the shopper clicked handbags to train a weighted auto-encoder. Then the 40 recommended handbags are ranked according to their anomaly scores. Fig. 5.5 shows the increase in the recall vs. the increase in the number of recommended handbags. The weighted AutoEncoder further improves the performance of handbag recommendation. It is noted that the weighted AutoEncoder performs well if the shopper-clicked handbags are visually similar, such like the data provided by subject #1.

To further evaluate the effectiveness of our proposed weighted AutoEncoder, we conduct an additional experiment in terms of outlier detection on a public image dataset: UIUC-Scene [8]. Follow the data preparation provided by [159], we use all 15 classes for UIUC-scene. For each class, we simulate outlier images with a proportion of 0.6 by sampling images from the other classes randomly. Then for each corrupted class, we train a weighted AutoEncoder and test the results on the

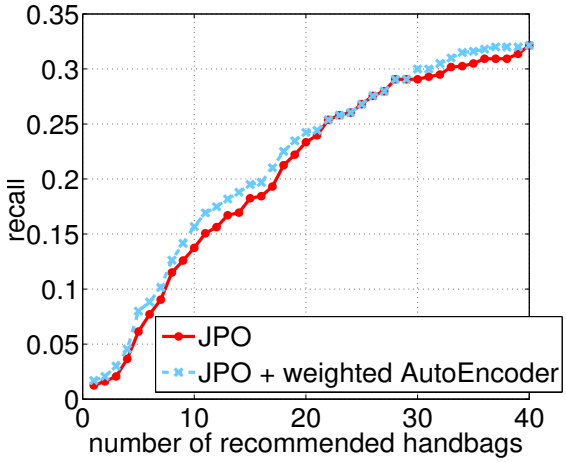


Figure 5.5: Comparisons of recall curves by varying the number of returned handbags over all 16 subjects for testing.

Table 5.1: Comparison results for outlier detection (60% outliers) on UIUC-Scene [8] datasets.

Method	mAP
PCA	0.5639
Kernel Density Estimation [156]	0.5995
Robust Kernel Density Estimation [168]	0.6082
One-Class SVM [150]	0.7737
Deep AutoEncoder [2]	0.7881
UOCL [159]	0.8157
Weighted AutoEncoder (proposed)	**0.9077**

same data by computing the anomaly score. The mean average precision (mAP) of all 15 classes are compared and summarized in Table 5.1.

5.5 Summary

In this chapter, we address the image-based handbag recommendation problem based on one-class SVM. We treat each shopper's clicked handbags as positive

data and analyze the content of those handbag images. As shoppers tend to prefer handbags with similar or the same attributes (e.g., color, pattern or style) for a certain period of time, we propose a joint learning to learn the attribute projection and one-class SVM classification together. This joint learning enables the projection of high-dimensional feature space into lower dimensional attribute space. The experimental results show that our method is promising on handbag recommendation.

Other than that, we propose a post-processing technique, named weighted AutoEncoder. It can detect the outliers in the recommendation list and achieve improvements for handbag recommendation. Weighted AutoEncoder assigns small weights to those dubious outliers and focuses more on the normal data reconstruction. Therefore, it attempts to deal with a high level of outliers. Experiments on UIUC-Scene verify that when the data is corrupted by a large fraction of outliers, weighted AutoEncoder achieves encouraging results in outlier detection.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Fashion recognition is an important problem in multimedia community. Handbag, as an indispensable item of a person's wardrobe, its related research is still limited. Imagine that a consumer is attracted to a handbag image on a fashion magazine or on an advertising board, and he/she desires more information about the handbag. In this case, a handbag recognition system can offer great convenience for the consumer. With the recognized model name, the consumer can search for more information online easily. Manufacturers can also benefit from this kind of system for the branding purpose. The function of recommendation is an important part of online shopping. To build up a handbag recommendation system and help shoppers find their preferred handbags will greatly enhance the user experience. This thesis focuses on handbag recognition and recommendation, which are important steps for building up a practical multimedia system for online advertising and e-commerce. Different techniques are proposed on discriminative representation for handbag recognition and common properties extraction for handbag recommendation.

Handbag recognition is a special fine-grained object recognition task, which

aims to differentiate the subtle differences among highly similar object classes. Different from other fine-grained objects, handbags of a brand are designed with diverse styles (shape, pattern and texture), and those sharing the same style (called SSC) only differ in color. Some styles of handbags are different due to the small decorations on local parts and subtle texture. The intra-class color variances within a SSC are often large due to illumination changes. To this end, several novel discriminative representations for handbag style and color are proposed. For handbag style representation, we propose two supervised mid-level patch selection procedures to select discriminative patches, w.r.t. one individual SSC and pairwise SSCs, respectively. We also propose a low-level complementary feature, extracted from texture enhanced mid-level patches, to capture the subtle texture of mid-level patches. We are the first to incorporate α-image with those feature descriptors to improve the classification performance. For handbag color representation, dominant color features are extracted to handle the illumination changes. In order to evaluate the algorithms, we construct a dataset covering about 10,000 images from 401 handbag models of two brands. A comparison is performed between our approach and two public available discriminative patch discovery methods [4, 30], which shows that the proposed patch selection achieves much higher accuracy (over 10%) for handbag recognition. We also compare the proposed framework with another fine-grained object recognition system. Our proposed method achieves over 10% improvement in classification accuracy. Time complexity of training and testing for the proposed framework is further reported and discussed, which shows that our method is able to be applied in real time. Additional experiments on eight image classification benchmarks show the improvements of incorporating complementary feature for object recognition tasks.

CNN is promising in many computer vision tasks. This motivates us to de-

sign a handbag recognition framework based on CNN. Previous CNN models do not embed the discriminative color information during training, and only use the ground truth class label for classification tasks. In order to deal with the aforementioned difficulties in small inter-class difference and large intra-color variation, in the deep CNN level, we propose a Feature Selective joint Classification-Regression CNN (FSCR-CNN) model. In this model, we incorporate discriminative color information to classify color sensitive objects. A distribution measuring how similar this class is to other classes is also assigned, to facilitate the classifier modeling for visually similar objects. In the system level, we design an end-to-end framework to recognize the model for a given input handbag image. It includes several steps. (1) Extracting the handbag regions by incorporating the symmetry property of the handbag. (2) Feeding the extracted proposals into the CNN detection model and the FSCR-CNN classification model separately. (3) Combining the detection scores and classification scores by employing a conditional probability model. The experiments show that the FSCR-CNN classification model outperforms CNN for fine-grained object recognition (8% improvement on handbags, over 5% improvement for the Oxford Flowers, the Stanford Dogs and the UCSD-Birds dataset). We compare the proposed style-to-color discriminative representation (SCDR) with the proposed FSCR-CNN for handbag recognition. It shows that FSCR-CNN outperforms SCDR on the BrandBag datasets (with 3% improvement on FSCR-CNN-G).

Handbag recommendation is another important step for building up a practical fashion recognition system. No existing recommender system is focusing on handbag recommendation. It is also difficult to describe a variety of different styles of handbags (attributes) or shoppers' tastes. However, we find that shoppers usually prefer handbags with similar attributes during a certain period of time. In order

to extract common properties of preferred handbags clicked by each shopper, we analyze the content of clicked handbags. We then propose a Joint learning of attribute Projection and One-class SVM classification (JPO) model. The original features are projected onto an attribute space. Meanwhile a hyperball is learned to separate the projected attribute features from the origin. Experimental results show that the proposed JPO performs favorably based on initial subject testing. We further improve the results of handbag recommendation by a post-processing. A weighted AutoEncoder technique to remove the recommended handbags which deviate from the others is presented. Experimental results show that the weighted AutoEncoder can work well for outlier detection, even when the data contain large fraction of outliers.

6.2 Future Work

Though some techniques from different aspects have been proposed for handbag recognition and recommendation in this thesis, there are still many issues needed to be further addressed. Based on the presented work, some future research directions are briefly summarized as follows.

1. In this thesis, we evaluate the performance of our handbag recognition framework on two brands. In the future, more branded handbags or even backpacks, suitcases could be combined. The bags taken under different environments, such as on the street, with large occlusion or various angles can be also considered. It would be interesting to explore the possibility to develop a large scale branded bag dataset and different applications based on it.

2. We propose two techniques for handbag recognition: the SCDR framework and the FSCR-CNN classification model. SCDR explicitly fixes the spatial posi-

tions of handbag patches, it is not be able to deal with severe rotation (larger than 30° in our experience) of the handbag. An alternative is to use Thin Plate Spline algorithm [169] to match the edges of the target handbag to a standard one (without rotation or distortion) [170]. SCDR can handle geometric distortions. Handbag parts suffering from geometric distortions will not be selected as discriminative patches. FSCR-CNN classification model is general and has the ability to tolerate rotation and distortion. Once we have sufficient images with rotated and geometrically distorted handbags in the training set, we can retrain the model. Both of these two techniques have advantages and limitations. How to design a handbag recognition method that takes the advantages of them is still an open issue.

3. The handbag recommendation system proposed in this thesis is based on the assumption that shoppers always prefer handbags with similar attributes during a certain period of time. In the future, more complicated situations can be addressed. For example, the shopper has several different kinds of preferences of handbags. Moreover, a clustering technique can be designed before recommendation, or a combination of recommendation and clustering can be conducted simultaneously.

4. We extract low-level features such as SIFT and color feature for handbag recommendation. More semantic features regarding the user profiles can be considered, such as the shopper's gender, age, favorite colors, favorite super-stars, or other purchase history. It would be able to solve the cold start problem with shopper's profile, i.e., conduct recommendation even when the shopper does not click any handbag. Developing new techniques to take into account all the factors about the shopper would be an interesting topic.

5. The proposed techniques for handbag recognition and recommendation on other objects or other applications can be further explored. Since we have verified that the complementary feature and FSCR-CNN model can improve the recognition accuracies on other benchmarks, the potential of those techniques are needed to be further studied. For example, i) how to fit them into more complicated frameworks and ii) how to modify them to train CNN models on ImageNet, so that more reliable pre-trained models can be obtained for various tasks. It is worth exploring how the proposed recommendation system can be generalized to other object types (e.g., shoes and clothes).

6. Fashion recognition system requires different functions. Such as handbag retrieval, clothes matching, different accessories (bags, necklaces, etc.) matching. For a more powerful fashion recognition system, it is necessary to study other fashion items. Although there exist some commercial image search engines, how to develop a more powerful and fine search engine for consumers, manufacturers, online shops or e-commerce could be an interesting future research direction.

Author's Publications

Journal Papers

1. **Y. Wang**, S. Li, and A. Kot, "DeepBag: recognizing handbag models", *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2072-2083, 2015.

2. **Y. Wang**, S. Li, and A. Kot, "On branded handbag recognition", *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1869-1881, 2016.

Conference Papers

1. **Y. Wang**, S. Li, and A. Kot, "Joint learning for image-based handbag recommendation", *in proceedings of IEEE International Conference on Multimedia and Expo (ICME)* (oral presentation), 2015.

2. **Y. Wang**, S. Li, and A. Kot, "Quality guided handbag segmentation", *in proceedings of IEEE International Conference on Digital Signal Processing (DSP)*, 2015.

3. D. Shyam, **Y. Wang** and A. Kot, "Histogram-based fast text paragraph image detection", *in proceedings of IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, 2015.

4. **Y. Wang**, S. Li, and A. Kot, "Complementary feature extraction for branded

handbag recognition", *in proceedings of IEEE International Conference on Image Processing (ICIP)*, 2014.

5. **Y. Wang**, S. Li, and A. Kot, "Category-Separating Strategy for branded handbag recognition", *in proceedings of 6th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, 2014.

Bibliography

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[2] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks." *Science (New York, N.Y.)*, vol. 313, no. 5786, pp. 504–507, 2006.

[3] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision (ECCV)*, 2014.

[4] B. Yao, A. Khosla, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[5] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Computer Vision, Graphics Image Processing (ICVGIP). Sixth Indian Conference on*, 2008.

[6] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, June 2011.

[7] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

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

[9] B. Safadi, M. Sahuguet, and B. Huet, "When textual and visual information join forces for multimedia retrieval," in *Proceedings of International Conference on Multimedia Retrieval (ICMR)*, 2014.

[10] W. Yin, T. Mei, C. W. Chen, and S. Li, "Socialized mobile photography: Learning to photograph with social context via mobile devices," *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 184–200, 2014.

[11] A. Puri and T. Chen, Eds., *Multimedia Systems, Standards and Networks*, 1st ed. Marcel Dekker, Inc., 2000.

[12] "The trouble with fashion recognition apps," Retrieved 5 November 2014.

[13] T. L. Berg, A. C. Berg, and J. Shih, "Automatic attribute discovery and characterization from noisy web data," in *Proceedings of the 11th European conference on Computer vision (ECCV)*, 2010.

[14] S. Liu, J. Feng, C. Domokos, H. Xu, J. Huang, Z. Hu, and S. Yan, "Fashion parsing with weak color-category labels," *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 253–265, 2014.

[15] A. Kovashka, D. Parikh, and K. Grauman, "Whittlesearch: Interactive image search with relative attribute feedback," *International Journal of Computer Vision*, pp. 1–26, 2015.

[16] A. Kovashka and K. Grauman, "Discovering attribute shades of meaning with the crowd," *International Journal of Computer Vision*, vol. 114, no. 1, pp. 56–73, 2015.

[17] M. Group, "Handbags-uk-january2015," [Online] Available: http://reports.mintel.com/sinatra/oxygen/brochure/id=715698, 2015.

[18] S. Branson, G. V. Horn, P. Perona, and S. J. Belongie, "Improved bird species recognition using pose normalized deep convolutional nets," in *British Machine Vision Conference (BMVC)*, 2014.

[19] A. Angelova and S. Zhu, "Efficient object detection and segmentation for fine-grained recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[20] J. Deng, J. Krause, and L. Fei-Fei, "Fine-grained crowdsourcing for fine-grained recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[21] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

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

[23] E. Amolochitis, I. Christou, and Z.-H. Tan, "Implementing a commercial-strength parallel hybrid movie recommendation engine," *Intelligent Systems, IEEE*, vol. 29, no. 2, pp. 92–96, 2014.

[24] A. van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[25] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *ACM International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*, 2011.

[26] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, "Combining content-based and collaborative filters in an online newspaper," in *ACM SIGIR Workshop on Recommender Systems: Algorithms and Evaluation*, 1999.

[27] N. K. A. Saxina and V. Venkataraman, "Building an image-based shoe recommendation system," *Stanford Final Projects*, 2013.

[28] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan, "Hi, magic closet, tell me what to wear!" in *ACM International Conference on Multimedia (ACM MM)*, 2012.

[29] J. J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *ACM International Conference on Research and Development in Information Retrieval (ACM SIGIR)*, 2015.

[30] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *European Conference on Computer Vision (ECCV)*, 2012.

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

[32] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

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

[34] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[35] W. Shen, B. Wang, Y. Wang, X. Bai, and L. J. Latecki, "Face identification using reference-based features with message passing model," *Neurocomputing*, vol. 99, pp. 339–346, 2013.

[36] J.-M. Geusebroek, A. W. M. Smeulders, and J. van de Weijer, "Fast anisotropic gauss filtering," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 938–943, 2003.

[37] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision*, 2004.

[38] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *European Conference on Computer Vision (ECCV)*, 2010.

[39] C. Cortes and V. Vapnik, "Support-vector networks," *Maching Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[40] L. Breiman, "Random forests," *Maching Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[41] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, pp. 197–387, 2014.

[42] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision, (ECCV)*, 2014.

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

[44] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *International Conference on Learning Representations (ICLR)*, 2014.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *European Conference on Computer Vision (ECCV)*, 2014.

[46] C. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

[47] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference (BMVC)*, 2014.

[48] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[49] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik, "Recognition using regions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[50] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *International Conference on Machine Learning (ICML)*, 2008.

[51] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[52] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment," in *European Conference on Computer Vision, (ECCV)*, 2014.

[53] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[54] O. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, "Cats and dogs," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[55] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and L. Yang, "Pfid: Pittsburgh fast-food image dataset," in *IEEE International Conference on Image Processing (ICIP)*, 2009.

[56] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *IEEE Workshop on 3D Representation and Recognition (ICCV workshop)*, 2013.

[57] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," Tech. Rep., 2013.

[58] Z. Ge, C. McCool, C. Sanderson, and P. Corke, "Modelling local deep convolutional neural network features to improve Fine-Grained image classification," in *IEEE International Conference on Image Processing (ICIP)*, 2015.

[59] D. Lin, X. Shen, C. Lu, and J. Jia, "Deep lac: Deep localization, alignment and classification for fine-grained recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[60] J. Krause, H. Jin, J. Yang, and L. Fei-Fei, "Fine-grained recognition without part annotations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[61] K. J. Shih, A. Mallya, S. Singh, and D. Hoiem, "Part localization using multi-proposal consensus for fine-grained categorization," *CoRR*, vol. abs/1507.06332, 2015.

[62] R. Farrell, O. Oza, N. Zhang, V. Morariu, T. Darrell, and L. Davis, "Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance," in *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[63] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[64] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur, "Dog breed classification using part localization," in *European Conference on Computer Vision (ECCV)*, 2012.

[65] E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars, "Fine-grained categorization by alignments," in *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[66] E. Gavves, B. Fernando, C. Snoek, A. Smeulders, and T. Tuytelaars, "Local alignments for fine-grained categorization," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 191–212, 2015.

[67] N. Zhang, R. Farrell, F. Iandola, and T. Darrell, "Deformable part descriptors for fine-grained recognition and attribute prediction," in *IEEE International Conference on Computer Vision (ICCV)*, 2013.

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

[69] K. Duan, D. Parikh, D. Crandall, and K. Grauman, "Discovering localized attributes for fine-grained recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[70] J. Krause, T. Gebru, J. Deng, L. Li, and L. Fei-Fei, "Learning features and parts for fine-grained recognition," in *International Conference on Pattern Recognition (ICPR)*, 2014.

[71] L. Zhang, Y. Yang, and R. Zimmermann, "Fine-grained image categorization by localizing tiny object parts from unannotated images," in *ACM International Conference on Multimedia Retrieval (ICMR)*, 2015.

[72] X. Wang, T. Yang, G. Chen, and Y. Lin, "Object-centric sampling for fine-grained image classification," *CoRR*, vol. abs/1412.3161, 2014.

[73] A. Angelova and P. M. Long, "Benchmarking large-scale fine-grained categorization," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[74] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie, "Visual recognition with humans in the loop," in *European Conference on Computer Vision (ECCV)*, 2010.

[75] A. Rejeb Sfar, N. Boujemaa, and D. Geman, "Confidence sets for fine-grained categorization and plant species identification," *International Journal of Computer Vision*, pp. 1–21, 2014.

[76] G. Research, "Movielens," [Online] Available: http://grouplens.org/datasets/movielens/, 2015.

[77] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *International Conference on Music Information Retrieval (ISMIR)*, 2011.

[78] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *International Conference on World Wide Web (WWW)*, 2005.

[79] C. Lu, W. J, Z. Guo, and R. Han, "User interest modeling based on terminal user behavior analysis," *Journal of Computational Information System 11: 1*, pp. 349–356, 2015.

[80] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2011.

[81] Q. Tu and L. Dong, "An intelligent personalized fashion recommendation system," in *International Conference on Communications, Circuits and Systems (ICCCAS)*, 2010.

[82] S. Ajmani, H. Ghosh, A. Mallik, and S. Chaudhury, "An ontology based personalized garment recommendation system," in *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, 2013.

[83] B. Kentner, *Color me a season: A complete guide to finding your best colors and how to use them.* Ken Kra Publishers, 1979.

[84] C.-M. Huang, C.-P. Wei, and Y.-C. Wang, "Active learning based clothing image recommendation with implicit user preferences," in *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2013.

[85] X. Yang, Y. Liu, Y. Guo, and H. Steck, "A Survey of Collaborative Filtering Based Social Recommender Systems," *Computer Communications*, vol. 41, pp. 1–10, 2014.

[86] L. Baltrunas and F. Ricci, "Experimental evaluation of context-dependent collaborative filtering using item splitting," *User Modeling and User-Adapted Interaction*, vol. 24, no. 1-2, pp. 7–34, 2014.

[87] P. Adamopoulos, "On discovering non-obvious recommendations: Using unexpectedness and neighborhood selection methods in collaborative filtering systems," in *ACM International Conference on Web Search and Data Mining (WSDM)*, 2014.

[88] K. Verstrepen and B. Goethals, "Unifying nearest neighbors collaborative filtering," in *ACM Conference on Recommender Systems (RecSys)*, 2014.

[89] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender sys-

tems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1273–1284, 2014.

[90] S. Kabbur and G. Karypis, "Nlmf: Nonlinear matrix factorization methods for top-n recommender systems," in *IEEE International Conference on Data Mining Workshop (ICDMW)*, 2014.

[91] Y. Wang, S. Li, and A. C. Kot, "Complementary feature extraction for branded handbag recognition," in *IEEE International Conference on Image Processing (ICIP)*, 2014.

[92] Y. Wang, S. Li, and A. C. Kot, "On branded handbag recognition," in *IEEE Transactions on Multimedia*, vol. 18, no. 9, 2016, pp. 1869–1881.

[93] W. Yin, J. Luo, and C. W. Chen, "Event-based semantic image adaptation for user-centric mobile display devices," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 432–442, 2011.

[94] G. Gualdi, A. Prati, and R. Cucchiara, "Video streaming for mobile video surveillance," *IEEE Transactions on Multimedia*, vol. 10, no. 6, pp. 1142–1154, 2008.

[95] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Fast image retrieval: Query pruning and early termination," *IEEE Transactions on Multimedia*, vol. 17, no. 5, pp. 648–659, 2015.

[96] Y.-H. Kuo, W.-H. Cheng, H.-T. Lin, and W. Hsu, "Unsupervised semantic feature discovery for image object retrieval and tag refinement," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1079–1090, 2012.

[97] Q. You, J. Yuan, J. Wang, P. Guo, and J. Luo, "Snap n' shop: Visual search-based mobile shopping made a breeze by machine and crowd intelligence," in *IEEE International Conference on Semantic Computing (ICSC)*, 2015.

[98] Y. Liu and T. Mei, "Optimizing visual search reranking via pairwise learning," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 280–291, 2011.

[99] R. Ji, L.-Y. Duan, J. Chen, L. Xie, H. Yao, and W. Gao, "Learning to distribute vocabulary indexing for scalable visual search," *IEEE Transactions on Multimedia*, vol. 15, no. 1, pp. 153–166, 2013.

[100] P. Li, M. Wang, J. Cheng, C. Xu, and H. Lu, "Spectral hashing with semantically consistent graph for image indexing," *IEEE Transactions on Multimedia*, vol. 15, no. 1, pp. 141–152, 2013.

[101] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan, "Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[102] L. Liu, J. Xing, S. Liu, H. Xu, X. Zhou, and S. Yan, "Wow! you are so beautiful today!" 2014.

[103] T. Chen, K.-H. Yap, and D. Zhang, "Discriminative soft bag-of-visual phrase for mobile landmark recognition," *Multimedia, IEEE Transactions on*, vol. 16, no. 3, pp. 612–622, April 2014.

[104] A. Angelova, S. Zhu, and Y. Lin, "Image segmentation for large-scale subcategory flower recognition," in *IEEE Workshop on Applications of Computer Vision (WACV)*, 2013.

[105] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. Lopez, and J. V. B. Soares, "Leafsnap: A computer vision system for automatic plant species identification," in *European Conference on Computer Vision (ECCV)*, 2012.

[106] N. Valliammal and S. Geethalakshmi, "Hybrid image segmentation algorithm for leaf recognition and characterization," in *International Conference on Process Automation, Control and Computing (PACC)*, 2011.

[107] Z. Zhao, L. Ma, Y. Cheung, X. Wu, Y. Y. Tang, and C. L. P. Chen, "Apleaf: An efficient android-based plant leaf identification system," *Neurocomputing*, vol. 151, pp. 1112–1119, 2015.

[108] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[109] J. Lim, C. Zitnick, and P. Dollar, "Sketch tokens: A learned mid-level representation for contour and object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[110] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[111] J. Shotton, R. B. Girshick, A. W. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake, "Efficient human pose estimation from single depth images," *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, vol. 35, no. 12, pp. 2821–2840, 2013.

[112] http://ufldl.stanford.edu/wiki/index.php/Softmax_Regression/.

[113] T. Stojic, I. Reljin, and B. Reljin, "Adaptation of multifractal analysis to segmentation of microcalcifications in digital mammograms," *Physica A: Statistical Mechanics and its Applications*, pp. 494–508, 2006.

[114] J. L. Véhel and P. Mignot, "Multifractal segmentation of images," *Fractals*, vol. 2, no. 3, pp. 371–378, 1994.

[115] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1512–1523, 2009.

[116] F. S. Khan, R. M. Anwer, J. van de Weije, A. D. Bagdanov, M. Vanrell, and A. M. Lopez, "Color attributes for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[117] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[118] A. Abdel-Hakim and A. Farag, "Csift: A sift descriptor with color invariant characteristics," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[119] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[120] L. jia Li, H. Su, L. Fei-fei, and E. P. Xing, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[121] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Comput. Vis. Image Underst.*, vol. 106, pp. 59–70, 2007.

[122] G. Griffin, A. Holub, and P. Perona, "Caltech-256 Object Category Dataset," Tech. Rep., 2007.

[123] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[124] L.-J. Li and F.-F. Li, "What, where and who? classifying events by scene and object recognition." in *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[125] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Institute of Technology, Tech. Rep., 2010.

[126] B. Yao, A. Khosla, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," http://vision.stanford.edu/discrim_rf/, 2011.

[127] Y. Wang, S. Li, and A. C. Kot, "Deepbag: Recognizing handbag models," in *IEEE Transactions on Multimedia*, vol. 17, no. 11, 2015, pp. 2072–2083.

[128] S. Yang, L. Bo, J. Wang, and L. G. Shapiro, "Unsupervised template learning for fine-grained object recognition," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[129] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," *CoRR*, vol. abs/1403.6382, 2014. [Online]. Available: http://arxiv.org/abs/1403.6382

[130] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "Wordnet: An on-line lexical database," *International Journal of Lexicography*, vol. 3, pp. 235–244, 1990.

[131] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, , and D. Batra, "Reducing overfitting in deep networks by decorrelating representations," *International Conference on Learning Representations (ICLR)*, 2016.

[132] A. Liaw and M. Wiener, "Classification and Regression by random Forest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[133] W. Shen, K. Deng, X. Bai, T. Leyvand, B. Guo, and Z. Tu, "Exemplar-based human action pose correction and tagging," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 1784–1791.

[134] J. Uijlings, A. Smeulders, and R. Scha, "Real-time visual concept classification," *IEEE Transactions on Multimedia*, vol. 12, no. 7, pp. 665–681, 2010.

[135] X. Geng and Q. Zhao, "Label distribution learning," *CoRR*, vol. abs/1408.6027, 2014.

[136] A. Sironi, V. Lepetit, and P. Fua, "Multiscale centerline detection by learning a scale-space distance transform," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[137] S. Godbole, S. Sarawagi, and S. Chakrabarti, "Scaling multi-class support vector machines using inter-class confusion," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.

[138] Z. Yan, V. Jagadeesh, D. DeCoste, W. Di, and R. Piramuthu, "HD-CNN: hierarchical deep convolutional neural network for image classification," *CoRR*, vol. abs/1410.0736, 2014.

[139] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2189–2202, 2012.

[140] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, 2013.

[141] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[142] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," *CoRR*, vol. abs/1412.4564, 2014.

[143] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, and X. Jiang, "Learning discriminative and shareable features for scene classification," in *European Conference on Computer Vision (ECCV)*, 2014.

[144] Z. Zuo, G. Wang, B. Shuai, L. Zhao, and Q. Yang, "Exemplar based deep discriminative and shareable feature learning for scene image classification," *Pattern Recognition*, vol. 48, pp. 3004–3015, 2015.

[145] J. Pu, Y.-G. Jiang, J. Wang, and X. Xue, "Which looks like which: Exploring inter-class relationships in fine-grained visual categorization," in *European Conference on Computer Vision (ECCV)*, 2014.

[146] Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *IEEE International Conference on Computer Vision*, 2013.

[147] Y. Wang, S. Li, and A. Kot, "Joint learning for image-based handbag recommendation," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2015.

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

[149] R. Pan, Y. Zhou, B. Cao, N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *IEEE International Conference on Data Mining (ICDM)*, 2008.

[150] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computing*, vol. 13, no. 7, pp. 1443–1471, 2001.

[151] L. Cui and Y. Shi, "A method based on one-class svm for news recommendation," *Procedia Computer Science*, vol. 31, pp. 281 – 290, 2014.

[152] S. Ben-David and M. Lindenbaum, "Learning distributions by their density levels: A paradigm for learning without a teacher," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 171 – 182, 1997.

[153] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed. World Scientific, 2006.

[154] D. M. Hawkins, *Identification of Outliers*, ser. Monographs on Statistics and Applied Probability. Springer Netherlands, 1980, no. 188.

[155] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," *ACM SIGMOD International Conference on Management of Data*, 2000.

[156] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. pp. 1065–1076, 1962.

[157] J. Kim and K. Grauman, "Shape sharing for object segmentation," in *European Conference on Computer Vision (ECCV)*, 2012.

[158] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," in *ACM SIGKDD Workshop on Outlier Detection and Description (ODD)*, 2013.

[159] W. Liu, G. Hua, and J. Smith, "Unsupervised one-class learning for automatic outlier removal," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[160] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computing*, vol. 10, no. 5, pp. 1299–1319, 1998.

[161] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis," *Journal of the ACM*, vol. 58, no. 3, pp. 11:1–11:37, 2011.

[162] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun, "Salient object detection by composition," in *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[163] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *International Conference on Machine Learning (ICML)*, 2006.

[164] Y.-F. Li, I. W. Tsang, J. T. Kwok, and Z.-H. Zhou, "Convex and scalable weakly labeled svms," *Journal of Machine Learning Research*, vol. 14, pp. 2151–2188, 2013.

[165] C. Christakou, L. Lefakis, S. Vrettos, and A. Stafylopatis, "A movie recommender system based on semi-supervised clustering," in *International Conference on Computational Intelligence for Modelling, Control and Automation, International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, 2005.

[166] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *International Conference on World Wide Web (WWW)*, 2001.

[167] M. Sugiyama and K. M. Borgwardt, "Rapid distance-based outlier detection via sampling," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[168] J. S. Kim and C. Scott, "Robust kernel density estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008.

[169] M. Powell, *A thin plate spline method for mapping curves into curves in two dimensions.* Computational Techniques and Applications (CTAC), 1995.

[170] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 24, pp. 509–521, 2002.