Feature representation and learning methods in visual search applications

Manandhar, Dipu

2019

Manandhar, D. (2019). Feature representation and learning methods in visual search applications. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/81689>

<https://doi.org/10.32657/10220/47995>

Feature Representation and Learning Methods in Visual Search Applications

Dipu Manandhar

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

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

04 April 2019

Date

A handwritten signature in black ink, appearing to read 'Dipu Manandhar', written over a horizontal line.

DIPU MANANDHAR

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

04 April 2019

Date

ASSOC. PROF. KIM-HUI YAP

Author Attribution Statement

This thesis contains material from 4 papers published in the following peer-reviewed journal / from papers accepted at conferences in which I am listed as an author.

A part of **Chapter 3** is published as Dipu Manandhar, Kim-Hui Yap, Zhenwei Miao, and Lap-Pui Chau. Lattice-support repetitive local feature detection for visual search. *Pattern Recognition Letters*, 98:123–129, 2017.

The contributions of the co-authors are as follows:

- I conducted all the experiments based on the developed idea and prepared the manuscript.
- Prof. Chau was involved in initial discussion of the project and ideas.
- Dr. Zhenwei gave useful feedback while developing the codes and helped to revise the manuscript.
- Prof. Yap closely supervised the entire work, provided insights on the idea, and gave suggestions for organization of manuscript and final submission.

A part of **Chapter 3** is published as Dipu Manandhar and Kim-Hui Yap. Feature repetitiveness similarity metrics in visual search. *IEEE Signal Processing Letters*, 24(9):1368–1372, 2017.

The contributions of the co-authors are as follows:

- I developed the idea, conducted all the experiments and prepared the manuscript.
- Prof. Yap closely supervised the entire work, provided useful insights on the idea, and gave suggestions for organization of manuscript, revision & final submission of the paper.

A part of **Chapter 4** is published as Dipu Manandhar, Kim Hui Yap, Muhammet Bastan, and Zhao Heng. Brand-aware fashion clothing search using cnn feature encoding and re-ranking. In *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*, pages 1–5. IEEE, 2018.

The contributions of the co-authors are as follows:

- I coordinated and collected the BrandFashion dataset, conducted all the experiments, and prepared the manuscript.
- Dr. Muhammet provided suggestion for collecting the dataset, and helped to develop the software tools used for image annotations.
- Mr. Zhao Heng helped in annotating the images.
- Prof. Yap supervised all the process, discussed the ideas, and provided suggestions to prepare the manuscript, and revised the paper.

A part of **Chapter 4** is published as Dipu Manandhar, Muhammet Bastan, and Kim-Hui Yap. Tiered deep similarity search for fashion. In *European Conference on Computer Vision*, pages 21–29. Springer, 2018.

The contributions of the co-authors are as follows:

- I developed the method, implemented the idea, conducted all the experiments and prepared the paper.
- Dr. Muhammet provided suggestions for implementation of idea and helped to revise the paper.
- Prof. Yap closely supervised the entire work, provided useful insights on the idea, and gave suggestions for organization of manuscript, revision and final submission of the paper.

04 April 2019

Date

DIPU MANANDHAR

To my Mother and Father.

Acknowledgement

I am deeply indebted to Assoc. Prof. Kim-Hui Yap, my PhD research supervisor, for his continual guidance, support and motivation. The research knowledge provided by him during the period is really invaluable. Besides research, the emotional support and care he provided during my studies was really commendable. I would like to thank him from the bottom of my heart.

I express my sincere thanks to Dr. Miao Zhenwei for his guidance and advice during the initial phase of my PhD studies. Most importantly, I would like to thank Dr. Muhammet Bastan for discussing the research ideas, sharing his knowledge and providing the support. I really appreciate his help and guidance. I am grateful to Dr. Sumit Bam Shrestha for his encouragement, motivation, and suggestions during my PhD studies.

I am grateful to my family and friends for all their support, motivation and encouragement for me to pursue my PhD. I would like to express my gratitude to ROSE Lab and her staffs for providing a sound working place, and providing the technical as well as administrative help. Many thanks to Dr. Dennis Sng, Mr. Joseph Lim and Ms. Wang Qian, the core management team of ROSE Lab. I am also thankful to all the wonderful friends and colleagues in the lab for their collaborations and friendship.

My sincere thanks to SINGA scholarship for providing me with financial assistance for my PhD and Nanyang Technological University for providing opportunity for me to complete my study. Finally, I would like to thank everyone who lend support to me and helped me during these years.

Abstract

This thesis studies and develops various image feature representation and learning methods for visual search applications. We study both handcrafted features as well as deep learning based representations. Handcrafted features based methods are light-weight and do not require large training data. However, they have been outperformed by deep learning methods in various vision-related problems in recent years. Nevertheless, deep learning methods generally require large data and computational power. In view of this, this thesis will study both handcrafted methods as well as deep learning methods for two selected domains, namely, visual landmark search and visual fashion search and application.

The first application develops algorithms for visual landmark search. The presence of repetitive patterns in landmark images causes *visual burstiness issue* which adversely affect the image representation. To tackle this, we propose a novel Lattice-Support Repetitive Local Feature Detection (LS-RLF) technique which first effectively detects repetitive patterns present in images and then uses the detection information during the image representation. As the repetitive pattern detection is early-vision problem and requires local features analysis, we develop our algorithms using handcrafted local feature-based representation. We also present a new Feature Repetitiveness Similarity (FRS) metric which quantize the repetitive and unique features independently and match them separately. The FRS metric makes use of information in repetitive patterns to enhance the search while avoiding the visual burstiness issue. Experiments conducted on three benchmark datasets namely, Oxford, Paris and Inria Holidays datasets show the effectiveness of the proposed methods.

The second application studies feature representation methods for fashion images using deep learning. We collect a new fashion dataset, NTUBrandFashion (NBF) dataset, with 10K fashion images which are richly annotated with essential elements of fashion: categories, attributes, and brand. We propose a new brand-aware fashion search (BAFS) which takes user brand preference into account. This search method uses a deep feature encoding which leverages on hierarchies of CNN activations to extract rich visual representation from clothing images. The brand-aware re-ranking in BAFS framework further improve the search performance. We also propose a new Attribute-Supervised Metric Learning (ASML) method to learn discriminative embedding from clothing images. This deep metric learning based method incorporates image attribute information to supervise the triplet

network training. This serves two purposes: (i) mining of informative triplets and (ii) treating the triplets in a soft-manner based on their importance, which helps in capturing similarity at different levels. Experiments conducted on NBF and DeepFashion datasets show the effectiveness of the proposed method.

The third work studies methods for fashion trend analysis and popularity prediction based on visual analysis of clothing images. We develop visual representation methods for fashion images while capturing their trend information to predict their popularity in terms of clickrates. We propose an image-based model and a sequence-based model using deep networks to predict the clickrate of the fashion items. The image-based model uses CNN network to predict the clothing popularity. The sequence-based method uses time-sequence of clothing images which uses CNNs to extract visual features and RNN to model the trend information. To the best of our knowledge, this is the first work to explore visual information for fashion forecasting of individual items. Experiments conducted on a dataset obtained from an online fashion company show promising results for fashion forecasting which outperform the recent comparative method.

Table of Contents

Acknowledgement	i
Abstract	iii
List of Figures	ix
List of Tables	xi
Abbreviations	xiii
1 Introduction	1
1.1 Motivation and Objectives	1
1.2 Main Contributions	5
1.3 Organization	6
2 Background and Literature Review	9
2.1 Visual Representation using Handcrafted Features	10
2.1.1 Handcrafted Local Features	10
2.1.2 Bag-of-Words (BoW) Model	11
2.2 Deep Learning-Based Visual Representation	14
2.2.1 Convolutional Neural Networks	15
2.2.2 Recurrent Neural Networks	17
2.2.3 Generative Adversarial Networks	19
2.2.4 Deep Feature Encoding and Deep Metric Learning	20
3 Visual Search with Repetitive Patterns	25
3.1 Introduction	25
3.2 LS-RLF Framework for Visual Search	29
3.2.1 Lattice-Support Repetitive Local Feature (LS-RLF) Detection	29
3.2.2 LS-RLF Image Representation and Retrieval	34
3.3 Feature Repetitiveness Similarity (FRS) Metric for Visual Search .	37
3.4 Experiments	40
3.4.1 Dataset and Evaluation	40
3.4.2 Experimental Settings	41
3.4.3 Qualitative Analysis of LS-RLF Detection	41
3.4.4 Performance of LS-RLF Visual Search	42

3.4.5	Ablation Studies: Effect of Weighting Factors	46
3.4.6	Performance of FRS-based Visual Search	48
3.5	Summary	50
4	Visual Fashion Search	51
4.1	Introduction	51
4.2	NTUBrandFashion (NBF) Dataset	53
4.2.1	Image Collection	54
4.2.2	Dataset Annotations	54
4.3	Brand-Aware Fashion Search (BAFS)	57
4.3.1	Joint Clothing and Logo Detection	58
4.3.2	PMAC Feature Extraction	59
4.3.3	Brand-Aware Re-ranking	60
4.4	Fashion Search using ASML	61
4.4.1	Overview of the ASML Framework	63
4.4.2	Attribute Prediction	64
4.4.3	Attribute Supervised Triplet Network	65
4.5	Experiments	67
4.5.1	Datasets	68
4.5.2	Experimental Settings	69
4.5.3	Evaluation Metrics	70
4.5.4	Results and Discussion: Proposed BAFS Framework	70
4.5.5	Results and Discussion: Proposed ASML Framework	74
4.6	Summary	78
5	Visual Fashion Forecasting	81
5.1	Introduction	81
5.2	Problem Formulation	83
5.3	Attribute Prediction	84
5.4	Visual Fashion Forecasting Models	85
5.4.1	Style-to-CR (Style2CR) Model	85
5.4.2	Image-to-CR (Im2CR) Model	88
5.4.3	Sequence-to-CR (Seq2CR) Model	89
5.5	Experiments	93
5.5.1	Dataset	93
5.5.2	Experimental Settings	94
5.5.3	Results and Discussion	95
5.6	Summary	97
6	Conclusion and Future Work	99
6.1	Conclusion	99
6.2	Future Work	101
	Author's Publications	103
	Bibliography	105

List of Figures

2.1	Key steps in BoW representation [5]	11
2.2	Alexnet architecture showing convolution, pooling, and fully connected layers. Image from [6].	16
2.3	A typical recurrent neural network and its unfolded version. The current hidden state is function of the current external input and the previous hidden state.	18
2.4	A typical LSTM cell with cell state (C_{t-1}), hidden state (h_t), forget gate (f_t), input gate (i_t), output gate (o_t). The hidden state and cell state in the next time-step are determined using interaction among these gates. Figure adapted from [7]	18
2.5	Block diagram of a typical Generative Adversarial Network (GAN).	20
3.1	Sample images with different types of repetitive patterns in man-made structures and natural sceneries.	26
3.2	Features that are quantized to the most frequent visual word for each image causing visual burstiness [8].	26
3.3	Examples of visual word matchings from repetitive patterns. Single visual word is matched to numerous features corresponding to the repetitive patterns.	27
3.4	Overview of the proposed LS-RLF framework in visual search	30
3.5	Illustration of repetitive pattern detection and clustering. (a) Image with overlaid detected features. (b) Image with dominant features (nodes) and connections (edges) between them. (c) Connected features after analysis. Different labels/symbols represent different repetitive feature types.	31
3.6	Repetitiveness weighting function with $\alpha = 0.5$ and $\mu = 10$	35
3.7	(a) Detected features and (b) Saliency Map overlaid on the image.	36
3.8	The proposed framework of visual search using Feature Repetitiveness Similarity (FRS) metric	39
3.9	Example 1: Repetitive pattern detection. The image on the top row (a) is obtained using the proposed LS-RLF method, the bottom row (d) is based on the method in [9]. Different groups of repetitive patterns are shown with different colors and annotated labels. Undetected but recovered new features are shown as square keypoints. See Section 3.4.3 for more explanations.	43

3.10	Example 2: Repetitive pattern detection on an example image. The image on the top row (a) is obtained using the proposed LS-RLF method, the bottom row (d) is based on the method in [9]. See Section 3.4.3 for more explanations.	44
3.11	Qualitative retrieval results using the proposed LS-RLF framework. Left-most column shows query images with the bounding box and the corresponding rows show the top 10 retrieved images.	46
3.12	Example queries images from the Oxford (top row) and Paris (bottom row) datasets with Average Precision	49
4.1	Sample images from NBF dataset with annotations.	54
4.2	Sample images from the NBF dataset.	55
4.3	Sample logos of 15 brands in NTUBrandFashion dataset. (a) Adidas, (b) All Star Converse, (c) Abercrombie and Fitch (d) Ferrari, (e) Kappa, (f) Levis, (g) New Balance, (h) Nike, (i) The North Face, (j) Playboy, (k) Puma, (l) Ralph Lauren, (m) Reebok, (n) Vans, and (o) Versace	56
4.4	(a) Screenshot of the developed annotation tool. (b) Sample fashion images annotated with clothing categories and brand logos with corresponding bounding boxes.	56
4.5	Sample fashion images representing various types of 'pattern' attributes.	57
4.6	Example of branded clothing search. Relevant and irrelevant retrieved images are shown with green border and red border respectively	58
4.7	Proposed Brand-Aware Fashion Search (BAFS) framework	59
4.8	Illustration of the desired feature space for the visual search and recommendation. Example clothing images exhibit similarities at different granularities represented by the concentric circles. The distances between the reference image (R) and images (A)-(E) reflect different degrees of visual similarity, with respect to image (R); visually similar images are closer while dissimilar images lie farther apart.	62
4.9	Architecture of the proposed attribute-supervised metric learning (ASML) network.	63
4.10	Sample detection of clothing and logo in images from NBF dataset. Detected items are indicated with bounding boxes with their confidence scores. Note that the detector can jointly detect both brand logos and clothing items under various deformations, background clutter, scales and orientations.	72
4.11	Qualitative comparison of retrieval results of various methods. Retrieved relevant and irrelevant images are shown with green border and red border respectively.	75

4.12	Sample qualitative results with query images and their respective top-5 retrieved images with WTBI [10], FashionNet [11] and our proposed method. Exact same instance matches are highlighted with green borders. Best viewed in color.	78
4.13	Retrieval examples of the proposed method from the DeepFashion dataset. Images in the first column are query images with their top-5 retrieved images in the respective rows. Correctly retrieved images from the same product ID are highlighted with green borders.	79
4.14	Retrieval examples of the proposed from the NTUBrandFashion dataset. Images in the first column are query images with their top-5 retrieved images in the respective rows. Correctly retrieved images of the same instances are highlighted with green borders. . .	80
5.1	Example screen-shot from <i>Amazon</i> online shopping site.	84
5.2	Overview of different forecasting models.	86
5.3	Sample images from <i>Pinterest</i> website. Fashion images in the top row have 15 times more <i>likes</i> compared to the fashion images in the bottom row [12].	88
5.4	Proposed Im2CR model for clickrate prediction	89
5.5	Histogram of number of items with respect to the number of the months the clickrates are available. Most the items only sold for once or twice.	90
5.6	Example clusters (each row represents a cluster) and representative fashion images obtained for the proposed sequence generation method in Seq2CR model.	91
5.7	Proposed Sequence2CR model for clickrate prediction.	92
5.8	Attribute prediction results on example fashion images. Correctly predicted attributes are highlighted in green.	95

List of Tables

2.1	Similarity metrics and formulas	14
3.1	Performance comparison of the proposed LS-RLF framework on Oxford, Paris, and Holidays dataset.	45
3.2	Ablation Study: Impact of Different Weighing Factors on Retrieval Performance of LS-RLF framework	47
3.3	Performance comparison of the proposed FRS metric on Oxford, Paris, and Holidays dataset.	48
4.1	List of 16 clothing categories in the NBF dataset	55
4.2	List of semantic attributes in NBF dataset	57
4.3	Average Precision (%) of logo and clothing detection.	73
4.4	Retrieval performance (mAP %) Vs. convolutional layers used for feature encoding	73
4.5	Retrieval performances of the proposed BAFS framework	74
4.6	Comparison of the proposed method BAFS with other comparative methods	75
4.7	Performance comparison of the proposed ASML framework for NTUBrand-Fashion dataset	76
4.8	Performance comparison of the proposed ASML framework for Deep-Fashion dataset.	77
5.1	Overview of the dataset used for fashion forecasting.	94
5.2	Performance of clickrate prediction for various methods on t-shirt, shirt and dress category.	96

Abbreviations

AI	Artificial Intelligence
AKM	Approximate K-Means
AQE	Average Query Expansion
ASML	Attribute Supervised Metric Learning
CNNs	Convolutional Neural Networks
DoG	Difference-of-Gaussian
FRS	Feature Repetitiveness Similarity
GBVS	Graph-based Visual Saliency
GPU	Graphics Processing Unit
HKM	Hierarchical K-Means
LS-RLF	Lattice-Support Repetitive Local Feature
LSTM	Long-Short Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
NN	Nearest Neighbors
QE	Query Expansion
RMAC	Regional Maximum Activation of Convolutions
SIFT	Scale Invariant Feature Transform
VV	Visual Vocabulary

Chapter 1

Introduction

1.1 Motivation and Objectives

Recent advancements in Artificial Intelligence (AI) and computer vision have enabled machines to perform vision-related tasks such as recognition, retrieval, and search. Visual representation is the key step in such tasks. This thesis studies and develops techniques for visual representation and feature learning for visual search and related applications.

The basic idea of visual representation is to encode the visual information in mathematical numbers or representation. Based on this idea, an image can be represented using statistical models of the pixel values. This global representations of images encode the whole image. However, they are not robust to occlusions, viewpoint variations, and local deformations. To handle this, handcrafted local features have been developed for image representation. Local features such as Scale Invariant Feature Transform (SIFT) [13], Speeded-Up Robust Features (SURF) [14] etc. are repeatable, robust to different deformations and illumination changes, and invariant to scale and orientation changes. Such local feature-based representation has been successfully used in image classification [15, 16], object detection [17, 18], retrieval and search [15, 19, 20]. The main advantage of local feature representation is that they do not require huge data and complex training strategies. More importantly, they can effectively represent local geometric information in images such as orientations and scales of the features.

While useful, the handcrafted features only extract image features based on fixed hand-engineered filters. For example, SIFT encodes features based on gradients

of the gray-scale images. Such handcrafted feature extractors have limited visual representation capability. In contrast, deep learning features learned using a large number of learnable filters have been shown to extract a wide range of representative and effective features for various tasks. In particular, using a large amount of data, Convolutional Neural Networks (CNNs) have outperformed the traditional local features based methods and demonstrated strong performance in vision-related tasks [21–25]. Nevertheless, deep learning methods generally require significant data and computational power during training. Further, they still experience limitations in certain issues such as inadequate handling of geometric consistency when performing image matching.

In view of this, this thesis will study both handcrafted methods as well as deep learning methods for two selected domains, namely, visual landmark search, and visual fashion search and applications.

Chapter 3 develops algorithms for visual landmark search which is a challenging problem in instance object recognition as it involves 3-D objects under different imaging conditions such as viewpoint and scale variation, occlusion, background clutters, etc. Most importantly, such application encounters the issue of repetitive features (e.g. windows, tiles, pillar, etc.) that poses additional difficulty (visual burstiness) in landmark recognition. We develop the visual search framework to tackle the burstiness issues due to repetitive patterns. As the problem requires low-level feature analysis and geometric information of the features to detect the repetitive patterns, we tackle the problem using handcrafted local features. We develop local-feature based image representations for visual landmark search which are robust to the presence of repetitive pattern in images.

Chapter 4 and Chapter 5 develop deep learning based feature representation for visual fashion clothing search and forecasting respectively. Both chapters deal with visual fashion domain and develop image feature representations for fashion clothing images. Fashion clothing analysis is an emerging research problem with huge commercial opportunities. Fashion analysis is a challenging domain as clothing is deformable, contains multiple attributes, and experiences various imaging conditions such as pose change, background clutters, etc.

In particular, Chapter 4 deals with visual fashion search. We study various features representation methods such as deep feature encoding from convolutional networks, deep metric learning to extract discriminative image embeddings from fashion images. Such representations are used for retrieving visually similar im-

ages to perform fashion search which have wide range of application in e-commerce and online shopping [26, 27] in various commercial platforms like eBay, Amazon, Taobao etc.

In Chapter 5 studies image representation methods for fashion images in visual fashion forecasting application. Similar to Chapter 4, we use deep networks such as convolutional networks to learn the features, and use them for predicting the popularity of fashion item in terms of clickrates. We study various methods of image representation: image-based, sequence-based, and latent-topic style representations. Different from the feature representation for fashion search in Chapter 4 which is independent of time, fashion forecasting deals with time-variant feature representation and takes the sales events into account. Nevertheless both applications aim to learn discriminative feature representation for fashion clothing images.

In the following, we explain the research problems studied in this thesis and clearly state our objectives.

- In the first work, we investigate the effect of repetitive patterns in visual search. Repetitive patterns provide a pleasing aesthetic to images and are ubiquitous in images from natural scenes as well as man-made structures (*e.g.* window facades, tiles, pillars etc). However, the repetitive patterns negatively affect the visual search performance. They tend to overwhelm the non-repeated features which cause visual burstiness issue [8], one of the major challenges in visual search. Thus, we aim to develop an effective image representation which is robust against the adverse effect of repetitive patterns. Previous works [8, 9, 28] directly downweight the contribution of repetitive features and only coarsely estimate repetitive patterns [9, 29] to handle the burstiness issue. As opposed to this, we aim to detect robust repetitive patterns in images and propose an image representation scheme which better handles the burstiness issue in visual search. Further, we will study the usefulness of the repetitive patterns for image matching. As repetitive pattern detection is an early-vision problem which requires low-level processing and local geometry analysis, we will use handcrafted local features to develop the algorithms. We will study them in landmark search applications where visual burstiness is a critical issue.
- In the second work, we study the methods of image representation and feature learning for visual search using recent deep learning techniques. We

apply the developed algorithms in visual fashion search which is a fascinating but challenging problem in computer vision. Fashion demonstrate diverse visual concepts with their designs, attributes, styles, and brands. The visual representation of a fashion image thus should capture the essential fashion concepts. We will first construct a new fashion dataset which captures the essential elements of fashion such as category, attribute, and brand. In terms of deep image representation, the existing methods are broadly categorized into two categories: (i) *deep features encoding* and (ii) *deep metric learning*. Previous deep features encoding [30–36] used features extracted from different CNN layers, but they do not exploit the hierarchy of information from different CNN layers. In view of this, we aim to develop a new deep encoding method which leverages on both low-level and high-level abstractions to extract rich features required for the search. On the other hand, existing deep metric learning based methods [10, 11, 37–40] learn embeddings in an end-to-end manner using a ranking loss (e.g. *triplet loss*). However, these methods treat all the training samples with equal importance which restrict learning the visual similarity at multiple granularities. Hence, we aim to develop a deep metric learning that captures image similarities at various granularities which is essential for fashion search and recommendation.

- The third work studies and develops the *time-variant visual representation* and apply it for the fashion popularity prediction. The main aim here is to learn the representation which captures the visual information of fashion images along with their trend (time) information. The fashion popularity prediction can assist fashion designers and companies to create new apparels, and manage their production and inventories. Hence, it has immense practical usefulness and commercial values. However, the problem is relatively new in computer vision and hence is not well studied. Recent work [41] represented fashion images by latent style representations and used traditional time series model for style forecasting. However, the method only predicts the popularity of a limited number of styles rather than individual items. In view of this, we aim to develop fashion popularity prediction models for individual item forecasting using deep convolutional and recurrent networks. We will develop an end-to-end trainable model for fashion forecasting, which jointly deals with visual information and time-series modeling, and predict the popularity.

1.2 Main Contributions

The main contributions of this thesis are highlighted below:

- (i) We propose a novel Lattice-Support Repetitive Local Feature (LS-RLF) detection method [1] to tackle the burstiness issue in visual search. The method detects repetitive patterns in images, estimate the underlying lattice structure, and recover the missing repetitive feature using Hough transform-based interpolation. We present an LS-RLF based image representation for subsequent visual image search. We also propose a Feature Repetitiveness Similarity (FRS) metric to tackle the overwhelming false matches on repetitive patterns. FRS metric matches repetitive and unique features separately. The FRS metric uses information from repetitive patterns to enhance the search performance while handling the burstiness issue. We demonstrate that repetitive features are not a nuisance but can be used to complement the unique features for visual search. Experiments conducted on three benchmark datasets in landmark search show the effectiveness of the proposed method.
- (ii) We collect a new NTUBrandFashion (NBF) dataset with rich annotations at different granularities such as brand, category, and attributes. The dataset captures essential visual concepts in fashion. We propose a brand-aware fashion search framework (BAFS) to demonstrate the importance of brand information in fashion search. The framework uses a deep feature encoding method to capture both low-level local information and high-level global abstractions from images. More importantly, we propose a new Attribute Supervised Metric Learning (ASML) to learn discriminative embeddings for fashion images. The proposed ASML method supervises the metric learning using attribute prediction information in a multitask network. The supervision guides two important aspects of metric learning: (i) selecting informative training samples, and (ii) treating the training samples based on their importance which helps to capture similarity at multiple granularities. We demonstrate the effectiveness of the proposed ASML method using two fashion datasets, namely NTUBrandFashion dataset and DeepFashion dataset
- (iii) We propose an image-based model (Im2CR) and a sequence-based model (Seq2CR) for fashion popularity prediction in terms of clickrates. The methods are end-to-end trainable and explore the visual information in fashion images using deep networks. We also present an extension to the recent

model in [41] to enable individual item forecasting. Experiments performed on the dataset obtained from an online fashion company demonstrated that both of the proposed methods outperform other comparative methods. Overall, we achieve promising results in fashion forecasting which can be useful for fashion companies to manage their sales and product inventories.

1.3 Organization

The organization for the rest of the thesis is given as follows.

In **Chapter 2**, we will provide a literature review and related backgrounds. We will introduce important terms and concepts in image feature representation. We will first present the methods of image representation using handcrafted features. Next, we will introduce deep learning and CNN, and explain feature representation and learning methods.

In **Chapter 3**, we will introduce Lattice-Support Local Repetitive Feature Detection (LS-RLF) algorithm for repetitive pattern detection. We will present a qualitative analysis of the detection algorithm. Next, we present an LS-RLF based image retrieval model to alleviate the visual burstiness issue. We also present a new Feature Repetitiveness Similarity (FRS) metric that first treats unique and repetitive features separately, and then use information in repetitive features to complement unique features to enhance the search performance.

In **Chapter 4**, we will introduce a new dataset for visual fashion search known as NTUBrandFashion. We will describe the key features of the dataset, the dataset collection and annotation procedure. We will then present the brand-aware visual search (BAFS) and show the importance of brand for fashion search and usefulness of the constructed dataset. The framework uses a new feature encoding method for visual search. Next, we will present a new Attribute-Supervised Metric Learning (ASML) method for learning image embedding. We will introduce a new architecture for supervising the metric learning based on attribute information using a multitask framework. We will explain the significance of providing supervision in training sample selection and learning similarity space at multiple granularities in visual fashion search. We will demonstrate the effectiveness of the proposed methods using two fashion datasets.

In **Chapter 5**, we introduce a challenging problem of visual fashion forecasting which has great practical usefulness. We present models for learning time-variant

feature representation for visual fashion popularity prediction. We study the problem of fashion popularity in terms of clickrate prediction. We will present an image-based model and a sequence based model for the clickrate prediction. We conduct experiments on a real-world dataset obtained from an online fashion company and present promising results for fashion forecasting which outperforms the existing comparative method.

Finally, in **Chapter 6**, we conclude the thesis and recommend future research directions.

Chapter 2

Background and Literature Review

The last two decades have witnessed dramatic developments in artificial intelligence, visual analysis, and information retrieval. There has been a huge paradigm shift in the techniques of visual analysis, from handcrafted feature representation to deep feature representation. The availability of big data, hardware computing resources like Graphics Processing Unit (GPU) and advancement in learning algorithms are the keys to the recent success of deep learning. Nevertheless, both handcrafted features and deep learning features have their own pros and cons. In this chapter, we will review the techniques of visual feature representation and learning using both handcrafted and deep learning methods.

This chapter is organized into two main sections. The first section reviews the techniques of visual representation using handcrafted features and its application for image retrieval. We provide a brief overview of local features, visual vocabulary, term weighting, and similarity metrics used in handcrafted feature-based visual search frameworks. In the next section, we explain deep learning and its use in visual representation and related applications. We provide a brief review of related works on image retrieval centered on deep learning. We will describe deep feature encoding and deep metric learning methods used in image representation.

2.1 Visual Representation using Handcrafted Features

The basic idea of visual representation using the handcrafted features is to encode the visual information into discrete representations and use them to compare the image similarities. Local features are often designed to be robust to occlusion, viewpoint variations, local deformations and photometric change. Once discriminative local features are extracted from images, various techniques/models such as Bag-of-Words (BoW), feature quantization, inverted indexing, geometric verification are applied to carry out the visual search.

2.1.1 Handcrafted Local Features

The first step in handcrafted feature-based visual search is to extract a collection of salient interest points or regions from images and encode them as local feature descriptors. Ideally, the extracted features should be *repeatable*, *precise* and *distinctive* such that image matching can be achieved effectively in the next step. Local feature extraction consists of two major steps, namely keypoint detection and extraction of feature descriptors.

Keypoints are interest points in images which carries useful information about the image contents. The keypoint detector can be *corner*, *blob* and *region* detectors. Harris detector [42] and SUSAN detector [43] are examples of corner detectors. Hessian detector [44], Laplacian-of-Gaussian (LoG) [45], Difference-of-Gaussian (DoG) [13], and Speeded Up Robust Feature (SURF) [14] are the blob detectors. Maximally Stable Extremal Regions (MSER) [46] is an example of region detectors.

Most of these feature detectors involve computation of derivatives on a large number of keypoints in images. In order to achieve the computational efficiency, DoG detector in SIFT [13] and SURF [14] have been developed. A detailed comparison of various keypoint detectors has been analyzed and discussed in [47]. SIFT is shown to achieve better performance under different imaging conditions [47].

Once the interest points or keypoints are detected, they are represented by descriptors which describe the image patch surrounding keypoints. Ideally, descriptors should be rotation-invariant, scale-invariant and robust to affine distortion and illumination changes. In literature, several descriptors have been proposed such as SIFT descriptor [13], Gradient Location and Orientation Histogram (GLOH) [48],

SURF [14], and Compressed Histogram of Gradients (CHoGs) [49]. The performance comparison of various descriptors in [48] showed that SIFT descriptors performed the best.

Although local features extracted from images provide a good representation of visual content, the process of feature matching is tedious and time-consuming. Therefore, Bag-of-Words (BoW) model is typically used for effective and efficient image matching based on local features. It is described in the following section.

2.1.2 Bag-of-Words (BoW) Model

The Bag-of-Words (BoW) model [16, 19, 20] has been popular in visual search applications in computer vision. The BoW model describes local features as discrete *visual words*, and an image as a histogram of visual words. Similarities between images are effectively computed using a similarity or distance metric between the BoW image histograms.

Figure 2.1: Key steps in BoW representation [5]

The BoW model involves the following steps: (i) detection of keypoints/interest regions, (ii) local features extraction, (iii) descriptors quantization into visual words to form the visual vocabulary and (iv) BoW representation i.e. finding the number of occurrences of each visual word found in an image (or forming a histogram of

word frequencies). Fig. 2.1 illustrates the main steps involved in BoW image representation model. The BoW image representations are compared using similarity metrics to obtain a rank list of similar images. The main steps in the BoW model are further elaborated in the following subsections.

A. Visual Vocabulary (VV) Generation

A *codebook* or *visual vocabulary* is generated for feature quantization in BoW representation. Generally, the vocabulary is obtained by performing K-means clustering on a large number of local features extracted from images. The visual vocabulary is used to quantize image features so that an image can be represented as a histogram of visual words.

Although the procedure is simple, the K-means algorithm becomes computationally expensive in a large corpus of features and a large value of K [20]. As a solution to this, the authors in [19] and [50] proposed Approximate K-Means (AKM) and Hierarchical K-Means (HKM) to obtain scalable vocabularies.

The Approximate K-means (AKM) [19] extends flat K-means by adopting approximate nearest neighbor search. As opposed to naive k-means where a significant amount of time is spent on searching the exact Nearest Neighbors (NN), AKM uses randomized k-d trees [51] to approximate the nearest neighbors. This significantly reduces the computational complexity and accelerate the visual vocabulary construction.

Similarly, the HKM [50] used a tree-structured feature quantization called *scalable vocabulary tree* (SVT). The main idea is to first apply K-means to the entire feature space and then recursively use K-means in the sub-space to obtain the finer partitions. Image features are quantized hierarchically through out the tree to reach the leaf nodes, which are treated as visual words. Compared to AKM, HKM has been shown to be efficient in constructing large vocabularies [50].

B. Inverted Indexing

The use of visual vocabulary offers an effective way to improve the search efficiency with the use of *inverted index* [52]. Each visual word stores ids of the images that contain the particular visual words. Hence, during the query stage, only visual word indexes that are present in the query image are checked. In practice, inverted indexing helps to save both computational time and memory usage.

C. Feature Assignment Strategies

In the BoW model, features are quantized and typically assigned to the nearest visual word in the vocabulary [19, 20, 50], which is referred to as a *hard assignment*. To improve the performance and reduce the impact in quantization loss, the author in [53, 54] proposed to use *soft-assignment* (SA) scheme, where a feature is assigned to few nearest visual words. In [53], a geometrically decreasing factor for k nearest visual words is proposed. Similarly, a descriptor-space soft assignment is proposed in [54], where features are assigned based on their distance to visual words. Another variant called *multiple assignment* (MA) is proposed in [55], where the descriptors are assigned to the k nearest visual words with equal weighting.

D. Term Weighting in BoW Representation

The BoW term weighting strategy is inspired from the standard *term frequency-inverse document frequency* (*tf-idf*) [56] in text retrieval. The popular (*tf-idf*) consists of two major factors: the local term frequency and global inverse document frequency. For a database with N images, the weight of the term w_{ij} for the visual word v_i in the database image D_j is given by,

$$w_{ij} = \frac{tf_{ij}}{tf_j} \log \frac{N}{N_i} \quad (2.1)$$

where tf_{ij} is the total number of visual word v_i in the database image D_j and N_i is the total number of images in the database containing visual word v_i . The term $\frac{tf_{ij}}{tf_j}$ indicates the visual word frequency and $\log \frac{N}{N_i}$ indicates the inverse document frequency.

E. Similarity Metrics

The BoW vectors for image representation are compared using distance metrics or similarity metrics. Generally, the vectors are first normalized. For a general *Minkowski* distance, the normalization factor related to L_k distance is given by (2.2),

$$n_j = \frac{1}{\|D_j\|_k} = \frac{1}{(\sum_{i=1}^z w_{ij}^k)^{\frac{1}{k}}} \quad (2.2)$$

where, $D_j = (w_{1j}, w_{1j}, \dots, w_{zj})$ is the BoW vector representation. The standard L_1 and L_2 correspond to $k = 1$ and 2 respectively in equation (2.2). A general expression for the Minkowski distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ is given by,

$$d_k(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n |x_i - y_i|^k \right)^{\frac{1}{k}}, \forall x, y \in \mathbb{R} \quad (2.3)$$

Table 2.1: Similarity metrics and formulas

Metrics	Formulas
Dot Product	$\text{dot}(\mathbf{x}, \mathbf{y}) = \sum_i x_i y_i$
Histogram Intersection	$\text{HI}(\mathbf{x}, \mathbf{y}) = \sum_i \min(x_i, y_i)$
Cosine Similarity	$\cos(\mathbf{x}, \mathbf{y}) = \frac{1}{\ \mathbf{x}\ _2 \ \mathbf{y}\ _2} \sum_i x_i y_i$
Bhattacharya's Coefficient	$\rho_{bhatt}(\mathbf{x}, \mathbf{y}) = \frac{1}{\sqrt{\ \mathbf{x}\ _1 \ \mathbf{y}\ _1}} \sum_i \sqrt{x_i y_i}$
Chi-square Distance	$\chi^2(\mathbf{x}, \mathbf{y}) = \sum_i \frac{(x_i - y_i)^2}{x_i + y_i}$

Besides, the $L1$ and $L2$ distance, various similarity metrics have been proposed and they are summarized in Table 2.1. In literature, the histogram intersection kernel and cosine similarity are widely used. The cosine similarity gives the similarity between images depending on the scalar product of $L2$ -normalized BoW vectors. The histogram intersection kernel, on the other hand, encodes the number of common bins in the BoW vectors.

F. Post-processing Strategies

The BoW model does not consider the spatial geometric information of the visual words. To perform geometric consistency check, methods such as *RANSAC* [57] and *Generalized Hough Transform* (GHT) [58] are used as post-processing steps. Spatial re-ranking [19, 20, 54] have shown to significantly improve the retrieval performance.

Another post-processing strategy is Query Expansion (QE) which is a form of relevance feedback and a technique used to enhance the retrieval performance in text retrieval [59]. Average Query Expansion (AQE) [60] used average tf-idf vectors of top- k retrieved images to refine the query. In [61], the authors improved the AQE by avoiding confusing features and learning the relevant spatial context. [62] proposes a method for fast formation of query expansion set using Hamming Embedding (HE) [63] while avoiding the computationally expensive geometric verification. The use of linear Support Vector Machine (SVM) to learn a discriminative weight vector for re-query has also been studied in [64].

2.2 Deep Learning-Based Visual Representation

In this section, we will present a review on deep learning based visual feature representation and learning approaches. The main advantage of deep learning features over handcrafted features is that deep features have a huge modeling capacity powered by a large number of learnable filter banks in the network. More

importantly, as the features are learnable, they show good performance over a wide range of domains and tasks. Successful training of such deep networks with large parameters has been possible due to the availability of big data and huge computing resources. As a result, there is a huge paradigm shift in visual feature representation and learning in recent year from hand-engineered visual features to end-to-end feature learning using deep learning. In this section, we will review feature encoding and metric learning using deep networks.

2.2.1 Convolutional Neural Networks

Deep learning [65] is a branch of machine learning which is typically characterized by multiple layers of computational units (e.g. neural networks). The layers learn multiple representations at different levels of abstractions in a supervised or unsupervised manner. Recently, deep learning has shown good performance on various tasks. Besides visual analytics, it has demonstrated strong performance in audio processing, signal processing, etc.

There exists a range of deep models such as convolutional neural networks (CNN) [6, 66], deep stacked auto-encoder, deep belief networks, deep Boltzmann machine. Among these, CNN is well suited for image data and has been widely used in visual analytics applications. The convolution operations in CNN can extract the information from highly correlated spatial image data. The deep architectures with such convolutional layers have achieved state-of-the-art performances in vision tasks such as image classification [6, 67, 68], object detection [23, 69], image segmentation [70], retrieval tasks [30, 31], etc.

Many architectures for CNNs have been proposed recently to better learn the image features and improve the performance. One of the seminal works in CNN framework, LeNet-5 was presented by LeCun et al. in 1990, which was used for handwritten digit recognition. This network was trained using backpropagation algorithm [71]. However, this network does not achieve good performance on large-scale complex datasets. In 2012, Krizhevsky et al. [6] proposed Alexnet with deeper architecture than LeNet-5. It achieved state-of-the-art performance for image classification in ImageNet Challenge in 2012. With this success, many deeper CNN architectures have been proposed to improve the performance. Some of the notable architectures are ZFNet [72], VGGNet [24], GoogleNet [73], ResNet [67] and DenseNet [68]. It is observed that the trend of network design moved towards deeper architecture. For example, ResNet is about 8 times deeper than VGGNet

Figure 2.2: Alexnet architecture showing convolution, pooling, and fully connected layers. Image from [6].

and about 20 times deeper than Alexnet. Nevertheless, the basic components of all these CNN architectures are very similar. We will describe the basic CNN architecture in the following sections.

A typical CNN consists of three basic components namely, convolutional, pooling and fully connected (FC) layers. Figure 2.2 shows a typical Alexnet architecture.

A. Convolutional layers

The convolution layers in CNN learn different hierarchies of features from raw image data. Each convolutional layer has several convolution filters, which compute different feature maps for a given input image. A neuron in a feature map is linked to neighborhood pixels in the earlier layer, which is known as the receptive field of the filter. The output feature map is then passed through a non-linear activation function. These convolutional layers are used in cascade to form multiple layers.

Alexnet shown in Fig. 2.2 has 5 convolutional layers and accepts images of size $224 \times 224 \times 3$. It uses the filters with receptive fields of 11×11 , 5×5 , and 3×3 . Studies have shown that earlier layers of CNN capture the low-level visual information such as lines, contours etc., while the later layers effectively capture higher semantic information like object or parts of the objects [72].

B. Non-Linear Activation

Generally, the output of the convolutional layers or fully connected layers are fed to an activation function to introduce the non-linearity in the network. Sigmoid, Rectifier Linear Unit (ReLU) [74] and tanh are widely used activation functions. ReLU has been popular as it is a simple operation and has shown good performance in various tasks. Several advances have been proposed to improve ReLU such as

Leaky ReLU (LReLU) [75], Parametric ReLU (PReLU) [76], Exponential Linear Unit (ELU) [77].

C. Pooling Layers

The pooling layers reduce the size of the feature maps. These layers try to introduce translation-invariance and provide robustness against distortions. Usually, pooling layers are placed between the convolutional layers to down-sample the feature map for subsequent layers. Some typical pooling operations are average pooling and max pooling.

D. Fully Connected (FC) Layers

Fully-connected layers are stacked after several layers of convolution and pooling. They provide a higher abstraction of images and are used for the final prediction [24, 72]. The neurons in the fully connected layers are connected to all the neurons of the previous layers. Hence, a large portion of weight parameters is concentrated in the FC layers in most of the CNN architectures. The final output of FC layers is fed to an output layer. For example, a classification network would use a softmax layer as the final output layer. A proper loss function is chosen according to the nature of the task. Some of the popular loss functions are softmax loss, hinge loss etc. The weights of CNNs are trained to obtain the global optimum using techniques such as stochastic gradient descent (SGD).

2.2.2 Recurrent Neural Networks

Recurrent neural network (RNN) [78] is a type of neural networks for processing sequential/time-series data $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(\tau)}$ where the outputs are function of inputs as well as previous states of the network. RNNs are characterized by internal state that allows to propagate information across time. The same update rule is applied to inputs in each time step. Most of RNNs models are able to process sequences of variable lengths.

Fig. 2.3 shows a typical RNN structure. The left figure shows the circuit diagram where the black square indicates an interaction between the state at time t and the state at time $t + 1$. The right figure shows the unfolded computational graph of the RNN that runs till the length of the sequence. The hidden state works as the memory of the network and can be represented as

$$\mathbf{h}^{(t)} = f(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}; \theta).$$

Figure 2.3: A typical recurrent neural network and its unfolded version. The current hidden state is function of the current external input and the previous hidden state.

Figure 2.4: A typical LSTM cell with cell state (C_{t-1}), hidden state (h_t), forget gate (f_t), input gate (i_t), output gate (o_t). The hidden state and cell state in the next time-step are determined using interaction among these gates. Figure adapted from [7]

Generally, the hidden states are initialized with zeros. In [79], it is shown that the initialization of hidden states with small non-zero number could achieve better performance and stability of the network.

Theoretically, RNNs are able to capture long-term dependencies in data to learn the information in sequences. However, in practice, they suffer from vanishing gradient or exploding gradient problem due to very long chain of recurrent layers. In order to handle these issues, Hochreiter & Schmidhuber in 1997 proposed Long Short Term Memory (LSTM) networks [80]. LSTMs can explicitly handle long term dependencies and can remember information for very long periods of time.

LSTM replaces the traditional repeating modules of hidden units such as "tanh" or "sigmoid" by memory cells which controls inputs and outputs using its special gates. Thus, LSTMs are special types of RNNs where the repeating modules are these special LSTM memory cells. Fig. 2.4 shows an LSTM cell structure and presents a computational diagram of signal flows in its gates. The cell state C_t acts like highway for information flow across the time. The forget gate, input gate and cell gate take previous hidden state and input and produces the outputs f_t, g_t & $\tilde{C}_t$ respectively. Using these outputs, the current cell state is then updated following $C_t = f_t \odot C_{t-1} + i_t \odot g_t$. The output of output gate and current cell state then produces the hidden state at current time step. The LSTM networks have shown to achieve remarkable results in wide range of applications such as language modeling [81], image captioning [82, 83], speech recognition [84, 85], forecasting [86–88].

Another variant of RNN called Gated Recurrent Unit (GRU) has been proposed in [89]. Similar to LSTMs, GRUs are designed to tackle the issues of vanishing and exploding gradients problems in RNNs and have gating units as in LSTMs. However, GRU merges the cell state and hidden state, and also combines forget gate and input gate into a single update gate. Hence, the architecture of GRU cell is simpler than LSTM cell. Detailed survey on RNNs and its variants has been carried out in [90, 91].

2.2.3 Generative Adversarial Networks

Generative adversarial networks (GANs) [92] are a class of neural networks which aims to learn a target data distribution in an unsupervised manner. GAN is initially proposed by Ian Goodfellow in 2014. Once trained, GANs can be used to generate artificial data (e.g. images) which follow the same distribution as the real training data.

A typical block diagram of GAN is show in Fig. 2.5. It consists of two distinct networks namely a generator network (G) and a discriminator network (D). The generator and discriminator networks can be standard multilayer perceptrons or CNN networks. The generator network takes some input (random noise) and outputs a 'fake' images which look likes real images. The discriminator network takes a image input and outputs whether the input is a fake image generated by the generator network or a real image from training data. The idea is to train both networks simultaneously where the generator tries to generate better images such

Figure 2.5: Block diagram of a typical Generative Adversarial Network (GAN).

that the discriminator network can not detect it as fake, while the discriminator network constantly tries to become a better detective to correctly classify the real and fake images. This adversarial training leads to an equilibrium state where the generator produces images from training data distribution and the discriminator is 50% confident that the generator output is real or fake.

GANs are used for generating artificial images including high resolution images in [93–95]. In recent years, GANs have been widely adopted for various applications such as image-to-image translation for domain transfer [96–98], generating images from text [99, 100], emoji generation and anime character creation [101, 102], image super-resolution [103], speech recognition [104].

2.2.4 Deep Feature Encoding and Deep Metric Learning

Discriminative feature representation is crucial in visual search and retrieval. In this section, we will discuss methods of feature representation and learning for visual search using deep learning. Broadly, they can be classified into two types; (i) deep features encoding, and (ii) deep metric learning. The former deals with feature encoding from different layers of CNN. The latter refers to an end-to-end approach to learn feature embedding space for image representation. The following sections will describe these two types of visual feature representations.

A. Feature Encoding using CNN

In this section, we will describe various methods to extract and encode features from various layers of a CNN network. The features can be extracted using the activations in the convolutional layers or from the fully-connected layers. CNNs trained for large-scale visual tasks such as Imagenet classification have shown to

capture generic visual features which can be adapted to various tasks [31, 105]. Razavian et al. in [31] demonstrated that off-the-shelf features from a CNN perform well in a wide range of visual recognition tasks including classification, scene recognition, fine-grained recognition, attribute detection and image retrieval. Inspired by this, several studies on CNN feature extraction [33–36] have been carried out. A good study of CNN feature representation and its related factors for visual recognition is presented in [33].

Several works have tried to extract CNN features from multiple image scales. The multi-scale orderless pooling [106] extracted features from the fully-connected layer using different image scales. Similarly, [36] divided the image into several square patches, and CNN features are extracted from them. In the similar spirit, Regional Maximum Activation of Convolutions (RMAC) [35] proposed to pool (max-pooling operation) the CNN activations from several square patches from images.

Researchers have also tried to treat the convolutional features as local features. The works in [107] and [34] used 3D activations in CNN and treated the vectors in the filter dimension in each spatial location as a local feature. Following this, images were encoded using the BoW model and VLAD descriptor respectively. Patch-CKN descriptors proposed in [108] used local feature detector, followed by patch encoding using convolutional descriptors. The work in [109] and [110] exploited object proposals based models for visual instance search. However, these methods are expensive as they can be a large number of proposals/interest points/square sub-patches from a single image. Moreover, such features extracted from CNN activations and fully-connected layers may not be discriminative for visual search. In the following, we will explain methods to learn discriminative features with deep metric learning.

B. Deep Metric Learning

The aim of deep metric learning is to extract meaningful embedding from images. The embedding should be such that visually similar images in image space should lie closer in the features space and vice-versa. In the early years, Bromley *et al.* [111] used deep Siamese networks with the contrastive loss for signature verification, which was extended to face verification [112] and dimension reduction [113]. Weinberger and Saul [114] used Large-Margin Nearest Neighbor (LMNN) metric learning for classification problems. These methods use contrastive loss, which minimizes the distance between the pair of the same class and keeps the negative pairs at a distance greater than a margin. Recently, the contrastive loss has been used for learning product similarity [115], matching user photo with online

products [116], face verification [117] etc.

Though successful, the contrastive loss imposes a constant margin and requires hard binary labels on similarities of training pairs, which may not be available in many practical cases. Different from this, the triplet constraint [118] requires relative similarities between images which is easier to estimate. Given a triplet $\{x^a, x^p, x^n\}$ consisting of an anchor image, a positive image and a negative image with corresponding embeddings $\{f_\theta(x^a), f_\theta(x^p), f_\theta(x^n)\}$, the triplet loss is given by

$$L_{tri}(\theta) = \left[\|f_\theta(x^a) - f_\theta(x^p)\|_2^2 - \|f_\theta(x^a) - f_\theta(x^n)\|_2^2 + m \right]_+, \quad (2.4)$$

where $[z]_+ = \max(0, z)$ is the hinge loss, m is the margin parameter and θ represents network parameters. This objective function pulls the similar data pairs $\{x^a, x^p\}$ closer in the feature space while pushing the dissimilar images $\{x^a, x^n\}$ apart. The triplet network has been widely used for various tasks such as FaceNet [37] for face recognition and verification, person re-identification [38, 119], fine-grained classification [120], image ranking [40], fashion search [10, 11, 26], etc.

Training with the triplet loss has been shown to be difficult as it requires mining of good (informative) training samples. Recently, several works have explored techniques for mining triplets for efficient learning. Song *et al.* [121] proposed lifted-structure feature embedding to take full advantage of examples in a batch, where all the positive and negative samples are mined to form the training triplets. In [37], the authors proposed semi-hard negative mining, where the negative training samples must lie inside a margin. In [122], the authors proposed in-triplet negative mining, where the positive and anchor are swapped to ensure the largest possible loss for the given triplet. Hermans *et al.* [123] recently proposed batch-hard training where all the possible triplets are mined and only hard examples are considered to compute the loss batch-wise rather than triplet-wise. This organization modification is only applicable when the exact same instance (for example, person ID) annotations are readily available. In many applications, such as image ranking [40], fashion search [26], the exact positive-pair annotation may not be available. For such cases, a reliable triplet selection is necessary. The work in [40] and [26] used additional features and manual verification for triplet selection, which is very costly.

Rather than training with a metric loss alone, simultaneous optimization of metric learning with auxiliary tasks in a multitask fashion has been developed. In [124] and [125], a combined contrastive loss and softmax loss is used for face identi-

fication and verification. Triplet loss and softmax loss have been used for face recognition [37] and fine-grained object recognition [126]. Lin *et al.* in [127] and Khamis *et al.* in [128] used attribute information and triplet loss respectively for person re-identification. Siamese network with the integration of semantic attributes has been used in [129] for street-to-shop clothing search. Liu *et al.* [11] proposed to jointly optimize classification, attribute predictions and triplet loss for fashion search. Most of these works use a weighted combination of multiple losses to update the network. Although the simultaneous optimization with multiple objectives has shown to learn better feature embedding, they lack the interactions among the multiple tasks.

Chapter 3

Visual Search with Repetitive Patterns in Landmark Recognition

3.1 Introduction

Repetitive patterns provide pleasing aesthetic to images, and are commonly encountered in images from natural scenes to man-made structures (see Figure 3.1). For example, in the city landscape, images often contain repetitive structures such as building facades, windows, tiles, etc. These repetitive patterns, however, creates negative impacts on the performance of visual search and recognition. These repeated elements tend to be highly similar and are likely to be quantized to the same visual word. Thus, the repetitive patterns cause *visual burstiness* issue [8, 9], one of the major challenges in visual search. In this chapter, we strive to achieve effective image representation which is robust against the adverse effect of the repetitive patterns in visual search.

The repetitive patterns pose significant challenges to image matching due to the following observations:

(i) Violation of Feature Independence Assumption: The repeated patterns in images violate the i.i.d. (independent and identically distribution) assumption required by state-of-the-art algorithms in image matching, retrieval, and reconstruction [8, 130, 131]. Hence, the presence of repetitive patterns results in inferior retrieval performance of the algorithms.

Figure 3.1: Sample images with different types of repetitive patterns in man-made structures and natural sceneries.

Figure 3.2: Features that are quantized to the most frequent visual word for each image causing visual burstiness [8].

(ii) **Burstiness of Visual Words:** Features from the same repetitive patterns are likely to be quantized into the same visual words, thereby causing visual words from these repeated patterns to overwhelm the non-repeated ones. This phenomenon is known as visual burstiness [8]. Figure 3.2 shows local features (hessian-affine) from repetitive elements which are quantized to the most bursty visual words. Due to the presence of such features, there can be undesirable feature matching between images. For example, in Figure 3.3(a), visual words from features in the right image match with a large number of repetitive features in the left image. Thus, the similarity score is dominated by the large number repeated features. This often leads to incorrect retrievals and degrades the performance of the visual search.

(a)

(b)

Figure 3.3: Examples of visual word matchings from repetitive patterns. Single visual word is matched to numerous features corresponding to the repetitive patterns.

(iii) Ambiguity in Geometric Matching: The spatial geometric verification of images are performed on the putative correspondences established during visual word matching. Hence, multiple matches of features from the same repetitive patterns will cause ambiguity in determining the true correspondences during geometric verification.

To handle these challenges, detection of the repetitive patterns is required prior to image matching. A number of repetitive pattern detection methods have been studied in the literature. Leung *et al.* [132] used local patches to find similar elements in the neighborhood. Park *et al.* [133] and Hays *et al.* [134] modeled the repetitive patterns with deformed 2D lattice. They used normalized cross-

correlation (NCC) on lattice template and vectors to find the dominant lattice structure in the image. However, these lattice-based detection methods [132–134] are not able to handle different types of repetitive patterns. Further, not all the repeated patterns can be represented by quadrilateral lattice elements. Most importantly, the detected lattice elements do not correspond to discriminative features, thus cannot be used for visual search/recognition directly. In image retrieval, repetitive building facades have been studied in [135] by measuring the similarity of motifs using NCC. Similarly, [136] employed a shift-invariant descriptor for lattice representation. However, both [136] and [135] can only handle a small dataset. This is because their methods do not produce discriminative features that are scalable with respect to the dataset dimension.

For scalable retrieval, the study on the effect of repetitive patterns has been done in [8, 9, 28]. Most of these methods try to down-weight the contribution of repetitive patterns in image representation. Jegou *et al.* in [8] first studied the burstiness phenomenon in images, where repetitive features vote multiple times for the same image. The authors proposed to use different down-weighting functions including $\log(\cdot)$, $\sqrt{\cdot}$ etc. on the term frequency of the visual words. Similarly, a generalized form of inverse document frequency called Lp-norm IDF [28] has been proposed to control the overwhelming effect of large components in the BoW image representation. The Lp-norm IDFs are negatively correlated with the term frequency and hence the large term frequencies are penalized during visual word weighting. Recently, Torii *et al.* proposed the ‘Reptile’ method [9] to handle the visual burstiness problem. The main idea is to detect the presence of repetitive patterns and then use soft-assignment strategy adaptively. The authors also suggested to directly truncate larger BoW histogram bins to reduce the impact of large terms in BoW vector.

The existing methods, however, either do not exploit repetitive pattern detection or only make a coarse estimation of repetitive patterns, thereby rendering only a sub-optimal solution to the problem. In view of this, this chapter presents a new Lattice-Support Repetitive Local Feature (LS-RLF) detection method to accurately detect different types of repetitive patterns in images. The proposed LS-RLF uses local feature and descriptor information that can be used directly for visual image search. Further, it estimates the underlying lattice structure, and extract missing features using Hough transform-based interpolation. The detection information is subsequently used for LS-RLF based image representation to handle the burstiness issue in visual search. In addition to this method, we will

also present a new Feature Repetitiveness Similarity (FRS) metrics that leverage on the repetitive patterns during similarity matching. The FRS metric handles unique and repetitive features separately, and fused them at the scoring stage. We demonstrate the effectiveness of the proposed methods in landmark search where visual burstiness due to repetitive patterns is a critical issue.

The rest of the chapter is organized as follows. We present our proposed LS-RLF visual search framework in Section 3.2. In Section 3.3, we present a new similarity metric which is based on the LS-RLF detection. Section 3.4 provides experimental results using the proposed methods. Finally, Section 3.5 provides a summary of the chapter.

3.2 Lattice-Support Repetitive Local Feature (LS-RLF) Framework for Visual Search

In this section, we present Lattice-Support-Repetitive Local Features (LS-RLF) based visual search framework. The main idea is to accurately detect the repetitive patterns in images, and then use the detection information for image representation. The flowchart of the proposed framework is shown in Figure 3.4. Overall, it consists of offline and online phases. In the offline phase, local features extracted from the database images are used for vocabulary construction and repetitive feature detection. Along with detection information, feature saliency information is incorporated to emphasize the foreground regions of interest and weaken the burstiness of repeated patterns in the backgrounds. The LS-RLF image representation is computed and stored in an inverted file. During the online phase, query images are passed through the same process as database images. The similarity between the query and the database images are computed, and the system returns the most similar images. The following sub-sections describe two main modules of the proposed framework, namely, LS-RLF detection, and image representation and retrieval.

3.2.1 Lattice-Support Repetitive Local Feature (LS-RLF) Detection

The LS-RLF detection consists of three major steps which are explained as follows.

A. Repetitive Feature Detection and Clustering

Figure 3.4: Overview of the proposed LS-RLF framework in visual search

Our studies from repetitive features indicate that they share many common attributes and have a similar scale, orientation and descriptor vector. These geometric as well as descriptor information can be used to measure the pairwise similarity between the features. In order to detect repetitive features and cluster them, we propose a set of similarity constraints (ζ) on the feature attributes as below:

1. Cosine similarity of central patch descriptors ($\Phi \in [0, 1]$)
2. Difference angle in orientations (θ *Radians*)
3. Ratio of relative scales (ρ)
4. Spatial proximity (L_2 -distance) between features (Δ)

Following extensive empirical evaluation, the following thresholds on the constraint ζ are selected: $\Phi \geq 0.9$, $|\theta| \leq 0.5$, $(1/1.3) \leq \rho \leq 1.3$, $\Delta \leq 10 \cdot \text{scale}$.

We formulate repetitive feature detection using an undirected graph where each feature represents a node in the graph. The edges are created if and only if the similarity between 2 nodes satisfies the constraints on ζ . Figure 3.5 illustrates the high-level idea of repetitive feature clustering. Figure 3.5(a) shows an image of facades and extracted local features overlaid on it. First, we create the linkages between similar features (as shown in Figure 3.5(b)) which satisfy the pairwise similarity required by ζ . For clarity, only the connections between dominant features are shown. By performing the connectivity analysis, features are clustered into different groups as shown by different labels in Figure 3.5(c). It can be noted

Figure 3.5: Illustration of repetitive pattern detection and clustering. (a) Image with overlaid detected features. (b) Image with dominant features (nodes) and connections (edges) between them. (c) Connected features after analysis. Different labels/symbols represent different repetitive feature types.

that there are a number of missing features which were not detected by well-known feature detectors. We also aim to recover those features using the proposed feature recovery technique described in the following section.

We use SIFT descriptors [13] to describe the local patches. As the descriptors from repeated pattern often experience some variations, direct use of the SIFT descriptors to measure the similarity between two repeated patches sometimes leads to erroneous results. We propose to use the central 2×2 SIFT patches which correspond to $2 \times 2 \times 8 = 32$ -dimensional feature descriptor. This is because the conventional SIFT descriptors are typically computed over an area larger than the scale of the detected keypoints. Hence, using the whole patch region for the repetitive pattern detection may include unwanted information in the repetitive features, especially for patterns that are closely spaced. Moreover, since the gradient information in the SIFT descriptors is Gaussian weighted with respect to the central feature location, the information of the descriptor is already dominated by

Algorithm 1 Repetitive Feature Detection and Clustering

Input: Feature frame $\mathcal{F} = (\mathbf{l}_i, s_i, o_i)$ and descriptor $\mathbf{d}_i$
Output: Clusters $\mathbf{C}$, each representing a type of pattern

- 1: Initialize Graph $\mathcal{G}_\zeta$ with ▷ Disconnected Graph
 $\mathcal{N}_p$ vertices i.e. features
 $\mathcal{E}$: Edges, $e_{ij} \in \mathcal{E} := \emptyset$
- 2: Extract central patch descriptor $\mathbf{d}_i$
- 3: Normalize descriptors $\mathbf{d}_i := \frac{\mathbf{d}_i}{\|\mathbf{d}_i\|}$ ▷ L_2 Normalization
- 4: **for all** features $\mathcal{F}$ **do**
- 5: Calculate constraint functions on ζ :
 $\Phi_{ij} = \mathbf{d}_i \cdot \mathbf{d}_j$; $\theta_{ij} = o_i - o_j$
 $\rho_{ij} = s_i/s_j$; $\Delta_{ij} = \|\mathbf{l}_i - \mathbf{l}_j\|_2$
- 6: **end for**
- 7: **for** $i = 1 : \mathcal{N}_p$ **do**
- 8: **for** $j = 1 : \mathcal{N}_p$ **do**
- 9: **if** $(\Phi_{ij} \geq 0.9 \ \& \ |\theta_{ij}| \leq 0.5 \ \& \ (1/1.3) \leq \rho_{ij} \leq 1.3 \ \& \ \Delta_{ij} \leq 10 \cdot (s_i + s_j))$ **then**
- 10: $e_{ij} = e_{ji} := \text{TRUE}$ ▷ Create Edge
- 11: **end if**
- 12: **end for**
- 13: **end for**
- 14: Compute the connected components $\mathbf{C}$ of $\mathcal{G}_\zeta$

the central regions. Our experimental studies show that the central patch is robust enough to determine the repetitive features when combined with other patch information such as scale, orientation, and location. A further advantage of directly using the feature descriptors to measure similarity is that it is not prone to quantization error. In addition, the use of orientation information in the proposed LS-RLF method helps to discriminate among different patterns.

Formally, the graph is constructed by linking similar repetitive features with edges. To capture the pairwise similarity, a binary affinity matrix is computed where elements (edges) are set to 1 if and only if all the 4 requirements by the constraint ζ are fulfilled. Similar to [9, 137], we extend the connected component analysis to group similar features and obtain initial feature clustering. The repetitive feature detection and clustering is summarized in Algorithm 1. Usually, only a few clusters contain a large number of cluster members which represent features from the repetitive patterns. Most of the remaining are singleton clusters which represent unique features in images. Thus, only a few dominant clusters are analyzed for further processing of repetitive patterns. As a result, the computational overhead is significantly reduced.

B. Cluster Refinement

Repetitive patterns in real-world images may experience variations due to differences in settings such as local occlusion, reflective surface, and viewpoint changes.

As a result, extracted feature descriptors from the same repetitive patterns may be grouped into different clusters. To address this issue, the proposed LS-RLF method develops a cluster refinement strategy to improve the initial clustering presented above.

The clustering of repetitive features is affected by local variations in images. The use of a stringent constraint ζ will differentiate different types of repetitive patterns but may split a class of repetitive feature into different clusters. Therefore, to reduce erroneous clustering, a stringent constraint (ζ) is first employed to achieve the initial clustering. Next, to merge the clusters containing the same type of repetitive features, the similarities between the clusters are measured using the inter-intra class distance ratio. The intra-class distance provides a measure of cluster tightness whereas inter-class distance measures the separation of clusters in the feature space. Hence, the similarity between two clusters c_i and c_j can be measured using inter-intra class distance ratio (R_{ij}) defined in (3.1),

$$R_{ij} = \frac{\|\bar{d}_i - \bar{d}_j\|_2^2}{\frac{1}{n_i n_j} \sum_{l=1}^{n_i} \|d_i^{(l)} - \bar{d}_i\|_2 \sum_{m=1}^{n_j} \|d_j^{(l)} - \bar{d}_j\|_2} \quad (3.1)$$

where $\bar{d}_i$ and $\bar{d}_j$ are the mean descriptors of clusters c_i and c_j , respectively and n_i and n_j are their respective cardinalities. Two clusters with ratio $R_{ij} < 1$ implies that they are similar in the descriptor space and therefore are merged into a single cluster. This refinement strategy addresses the misclassification of initial clustering.

C. Lattice Estimation and Feature Recovery

In this section, we aim to estimate underlying lattice structures of repeated features. The spatial occurrence of these repeated features is formulated with parametric models. In particular, Hough-transform is used to determine the parameters of different patterns and their lattice structures. The presence of large bins in Hough space refers to specific patterns in the image. Moreover, the method also aims at recovering the missing (undetected) features in the pattern. The lattice estimation and feature recovery technique in the LS-RLF framework is summarized in Algorithm 2.

Local feature locations are used to estimate the underlying lattice structures in each cluster c . First, the locations (x, y) in the image space are mapped to the Hough parametric space using Hough transform. Large bin counts $\{B_k\}$ in Hough accumulator cells indicate the presence of a specific structure in the image. To

Algorithm 2 Lattice Estimation & Feature Recovery

Input: Feature location (x_i, y_i) of repeated patterns
Output: Lattice pattern Λ^k , recovered features Υ

```

1: for all Clusters  $c$  do
2:   Get repeated feature locations  $(x_i, y_i)$ 
3:   Hough Space Mapping:  $(x_i, y_i) \rightarrow (a_i, b_i)$ 
4:   Find the large bins  $\{B_k\}$  in Hough accumulator cells.
5:   Compute hough-bin-pixels  $\{P_k\}$  for  $\{B_k\}$ 
6:   for all  $P_k = \{p_1^k, p_2^k, \dots, p_n^k\}$  do
7:     Estimate repetition interval  $\Lambda^k$  ▷ Lattice Pattern
8:     if Interval  $(p_i^k, p_{i+1}^k) \approx m \cdot \Lambda^k$  then
9:       Interpolate locations  $\kappa$  ▷ Missing Feature
10:    end if
11:  end for
12:  Compute descriptors  $\Upsilon$  on locations  $\kappa$ 
13:  if  $\Upsilon$  is valid then ▷ Repetitive Consistency Check
14:    Add  $\Upsilon$  to the cluster ▷ Recovered Valid Features
15:  end if
16: end for

```

find the corresponding set of points $\{P_k\}$, bins with large counts $\{B_k\}$ are back-projected to the image space. Specifically, points $P_k = \{p_1^k, p_2^k, \dots, p_n^k\}$ in the pattern repeat with an interval Λ . The undetected features in the image can then be predicted if the interval between two consecutive detected points is multiples of the interval Λ . Next, missing feature locations κ are interpolated where new descriptors Υ are computed to extract the undetected local repeated features. Finally, to ensure the repetitiveness consistency, recovered features Υ are validated by checking descriptor similarity using cosine measure.

3.2.2 LS-RLF Image Representation and Retrieval

In this section, we present the LS-RLF based image representation using the results obtained from repetitive pattern detection using LS-RLF detection algorithm.

The main reason for poor retrieval in other works is due to burstiness issue is overwhelming false matches on visual words from repetitive patterns. We aim to reduce such overwhelming matches to address the issue. Our LS-RLF based image representation $\mathbf{I}$ for an image with N features is given by,

$$\mathbf{I} = (w_1, w_2, \dots, w_i, \dots, w_z)^\top, \quad (3.2)$$

$$\text{where } w_i = \sum_{j=1}^N \sum_{k=1}^K w_R w_S w_{SA}(k) \cdot 1[v_j^{(k)} = i]. \quad (3.3)$$

Here $1[v_j^{(k)} = i]$ is the indicator function that is equal to 1 if the k^{th} nearest neigh-

Figure 3.6: Repetitiveness weighting function with $\alpha = 0.5$ and $\mu = 10$

bor of descriptor $\mathbf{d}_j$ is quantized to visual word index i . K is the total number of feature assignments and z is the total number of visual words. The most important aspect is that the designed visual word weight w_i integrates 3 factors in its formulation: repetitiveness score (w_R), saliency weighting (w_S) and soft-assignment weighting (w_{SA}). The significances of these factors are explained as follows:

(i) Repetitiveness Score (w_R)

The information from the detected repetitive patterns is incorporated into image representation and in subsequent image scoring. The cluster cardinality (n) of the detected features is an indicator of feature repetitiveness. The bigger the cluster, the more repetitive the feature is. Each feature f_i in the image is assigned a repetitiveness score $w_R \in (0, 1]$ based on the sigmoidal function in (3.4).

$$w_R = \frac{s}{1 + \exp(\alpha(n - \mu))} \quad (3.4)$$

The parameter s is the scaling factor, and parameters α and μ are positive real numbers that control the decay and shift of the sigmoid function.

Figure 3.6 shows the repetitiveness function which provides different weights to features with different repetitiveness. The repetitiveness score decreases with an increase in cluster size (n). The repetitiveness score ensures features with high repetitiveness are de-emphasized so that they do not overwhelm the BoW histogram. This will alleviate the burstiness issue due to the repeated patterns. The

Figure 3.7: (a) Detected features and (b) Saliency Map overlaid on the image.

values of the parameters α and μ in (3.4) are experimentally set to 0.5 and 10 respectively.

(ii) Saliency Weighting (w_S)

Saliency weighting is integrated into the overall weighting to emphasize the salient foreground object of interest, and de-emphasize the background bursty features such as trees, vegetations etc. The factor $w_S \in [0, 1]$ is the saliency weight extracted from the saliency map $S \in [0, 1]$ obtained using Graph-based Visual Saliency (GBVS) [138]. GBVS estimates a saliency map S using a graph in which the directed edge e from a point (i, j) to point (p, q) has weight

$$e((i, j), (p, q)) = \left| \log \frac{M(i, j)}{M(p, q)} \right| \exp \left(-\frac{(i - p)^2 + (j - q)^2}{2\sigma^2} \right), \quad (3.5)$$

where M is the feature map obtained using biologically inspired filters and σ is a free parameter. An activation map is generated by eigenvector computation based on graph theory, which is normalized and combined to obtain the final saliency map M_S . Figure 3.7 shows the local features and the saliency map overlaid on an image. Highlighted orange and yellow regions indicate informative foreground regions with high visual saliency. Bursty features from the background are suppressed while those from foreground region of interest are boosted. To obtain the weights w_S from the map S , a gamma transformation is used for saliency value calibration i.e. $w_S = M_S(x, y)^\gamma$, where (x, y) is the feature location. For all the experiments,

the value of γ is empirically set to 0.8 as it has been shown to achieve good performance.

(iii) Soft-assignment Strategy (w_{SA})

We extend the soft quantization strategy in [53]. The factor w_{SA} is set to $1/(2^{k-1})$ which represents the k^{th} soft-assigned weight. This provides geometrically decreasing weights with respect to k and helps to reduce the impact of quantization loss.

Let $\mathbf{Q}$ and $\mathbf{I}$ be the LS-RLF image representations for the query and database images. The similarity score between the query image $\mathbf{Q} = \{q_i\}_{i=1}^z$ and a database image $\mathbf{I} = \{w_i\}_{i=1}^z$ is calculated using cosine similarity between the representations,

$$Score(\mathbf{I}, \mathbf{Q}) = \frac{\sum_{i=1}^z w_i \cdot q_i \cdot idf(i)^2}{\|\mathbf{I}\|_2 \|\mathbf{Q}\|_2}, \quad (3.6)$$

where idf is the inverse document frequency of visual words. The framework returns the top-ranked similar images to the users.

3.3 Feature Repetitiveness Similarity (FRS) Metric for Visual Search

In the previous section, we presented an LS-RLF-based visual search framework. The main idea was to detect the repetitive patterns and represent images using the detected information to handle the burstiness issue. In this section, we will present an image similarity metric centered on the repetitive pattern detection.

The choice of metrics used for comparing image similarities can significantly affect the retrieval performance. In literature, the histogram intersection kernel and cosine similarity measure are widely used similarity metrics. Different similarity metrics have their own characteristics. For example, the cosine similarity gives the similarity between images based on the scalar product of L_2 -normalized BoW vectors. The histogram intersection kernel, on the other hand, encodes the number of maximum counts of matches instead of considering the entire set of features. Consider two images containing 2 and 5 counts of a particular visual word. For cosine similarity, the contribution through this visual word will be $2 \times 5 = 10$, ignoring IDF and normalization for clarity. However, the histogram intersection kernel gives maximal match counts, i.e. $\min(2, 5) = 2$. Hence, different similarity metrics can give different scores for the same pair of images. Therefore, designing

a good similarity metric is crucial in visual search.

In view of this, we present a new feature repetitiveness similarity (FRS) metric for visual search that utilizes information from repetitive pattern detection. Our key motivation is that repetitive patterns are significant cues for recognizing visual objects. To identify an image object, unique features (*i.e.* non-repetitive) may be sufficient to characterize it. However, if the image has a large number of repeated structures, they become important information that should be used to complement the unique features. Thus, we propose to handle repetitive features and unique (non-repetitive) ones separately *i.e.* we quantize and match these features independently such that bursty false matches between unique and repetitive features are avoided. Moreover, the proposed FRS metric tries to incorporate information from repetitive patterns to enhance the retrieval performance.

The overview of the FRS metric search framework is shown in Figure 3.8. Compared to the LS-RLF retrieval, the main idea here is to treat the unique and repetitive features separately, and fuse them at the scoring stage. The database images are first ranked based on the FRS metric, and then query expansion [60] is performed to return the top retrieved images for the users. The proposed framework uses the LS-RLF detection algorithm proposed in Section 3.2.1 to detect the repetitive patterns. The detected repetitive patterns contain useful information for image matching. Specifically, the cardinality of the detected cluster is a good indication of feature repetitiveness. A cluster with a large number of features refers to highly repeated structures whereas singleton cluster or cluster with very few members mean their features are unique in appearance.

Formally, let I be an image with $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{N_c}\}$ detected repetitive feature clusters where N_c is the total number of cluster detected. Let $n(\mathbf{c}_j)$ denotes the cardinality of the cluster $\mathbf{c}_j$. Each cluster $\mathbf{c}_j$ is identified as unique ($\mathcal{C}_U$) or repetitive ($\mathcal{C}_R$) based on the rule,

$$\mathbf{c}_j \in \begin{cases} \mathcal{C}_U & \text{if } n(\mathbf{c}_j) < T \\ \mathcal{C}_R & \text{if } n(\mathbf{c}_j) \geq T, \end{cases} \quad (3.7)$$

where T is a positive threshold empirically set to 5.

The features $F_U = \{f : f \in \mathcal{C}_U\}$ are interpreted as non-repetitive, and hence are called unique features, whereas features $F_R = \{f : f \in \mathcal{C}_R\}$ are called repetitive features. These 2 types of features are quantized separately using function ($\mathcal{Q}$)

Figure 3.8: The proposed framework of visual search using Feature Repetitiveness Similarity (FRS) metric

in (3.8) based on a codebook with visual words: $\{v_1, v_2, \dots, v_i \dots v_z\}$. As a result, two BoW histograms: unique $\mathbf{I}_{\mathcal{U}}$ and $\mathbf{I}_{\mathcal{R}}$ are obtained for each image.

Next, the score between the query image and the database image is computed as follows. Let w_i and q_i be the i -th visual word occurrence for $\mathbf{Q}$ and $\mathbf{I}$, respectively, and $idf(i)$ be associated inverse document frequency. The score between the query and database image is computed using similarity score (S) in (3.9) for both $\mathbf{I}_{\mathcal{U}}$ and $\mathbf{I}_{\mathcal{R}}$ representations.

$$\mathcal{Q}(f) = \arg \min_i \|f - v_i\|_2^2 \quad (3.8)$$

$$S_k(\mathbf{I}, \mathbf{Q}) = \frac{\sum_{i=1}^z w_i \cdot q_i \cdot idf(i)^2}{\|\mathbf{I}_k\| \|\mathbf{Q}_k\|}, \quad k \in \{\mathcal{U}, \mathcal{R}\} \quad (3.9)$$

$$\text{where, } \|\mathbf{I}\| = (\sum_i w_i^2)^{\frac{1}{2}} \text{ and } \|\mathbf{Q}\| = (\sum_i q_i^2)^{\frac{1}{2}}.$$

Let $S_{\mathcal{U}}$ and $S_{\mathcal{R}}$ be the scores for the unique and repetitive representations. The final similarity score is computed using (3.10) which takes the unique as well as repetitive features, and the degree of repetitiveness α into account,

$$S_{\text{FRS}} = \beta S_{\mathcal{U}} + \lambda \cdot \alpha S_{\mathcal{R}} \quad (3.10)$$

$$\text{where, } \alpha = \frac{n(\mathcal{C}_{\mathcal{R}})}{n(\mathcal{C}_{\mathcal{R}}) + n(\mathcal{C}_{\mathcal{U}})}, \text{ and } \beta = \frac{n(\mathcal{C}_{\mathcal{U}})}{n(\mathcal{C}_{\mathcal{R}}) + n(\mathcal{C}_{\mathcal{U}})} \quad (3.11)$$

Here $\beta = 1 - \alpha$ is a measure of feature uniqueness. λ is a regularization parameter experimentally set to 0.9 to calibrate the scores $S_{\mathcal{U}}$ and $S_{\mathcal{R}}$. The joint score S_{FRS} in (3.10) fuses the information of unique and repetitive features while alleviating the problem of bursty false matching.

It is noted that the degree of repetitiveness α and hence β in (3.11) are not design parameters, but rather their values are determined based on the outcome of repetitive pattern detection. In the absence of repetitive patterns, the repetitive term (α) becomes zero and hence all features will be treated equally.

3.4 Experiments

3.4.1 Dataset and Evaluation

The proposed methods are used to perform image retrieval and evaluated using three benchmark landmark datasets namely, Oxford, Paris and Inria Holidays datasets. These datasets have been extensively used by most of the well-known works in landmark image search and retrieval such as [21, 30, 31, 34, 54, 63, 139] and studied by survey papers [21, 140] and are considered as standard benchmark datasets for landmark image retrieval in computer vision community.

1. **Oxford dataset:** The Oxford dataset [19] contains 5,062 database images which are annotated with ground-truth for 55 query images from 11 distinct Oxford landmarks. This dataset is challenging as images are taken under significant variations of illumination, scale, and viewpoint.
2. **Paris dataset:** The Paris dataset [54] contains 6412 images of various Paris landmarks which are also annotated with labels for 55 query images.
3. **Inria holidays dataset:** The Inria Holidays dataset [63] is a set of personal holiday images which include a large variety of scenes with 500 query images in total.

For performance evaluation, Average Precision (AP) is used to calculate the area under the precision-recall curve. The APs for all the query images are averaged to give mean Average Precision (mAP) for the particular dataset.

3.4.2 Experimental Settings

For a fair comparison, we use the same dataset and experimental settings for all comparative methods and our proposed methods. We use Difference-of-Gaussians (DoG) function to extract the interest points which are then wrapped by affine elliptical regions resulting in DoG-Affine detector. The affine regions are described by 128-dimensional SIFT descriptor [13] followed by *RootSIFT* [64] normalization. Visual vocabularies (VV) are constructed with approximate K-means using FLANN library [141]. An inverted index file is constructed using the visual vocabulary where database images and their respective features are indexed. Algorithms are developed using MATLAB and VLFeat library [142] in a computer with i5-4570 CPU and 32GB memory.

In the following sections, we first provide a qualitative analysis of LS-RLF detection algorithm. Next, we present the retrieval performance of the proposed LS-RLF visual search. Then, we discuss the effect of different weighting factors in LS-RLF image retrieval. Finally, we present the performance of the proposed FRS metric visual search.

3.4.3 Qualitative Analysis of LS-RLF Detection

This section presents the qualitative results and analysis on repetitive pattern detection by the proposed LS-RLF method and compares with the state-of-the-art reptile method in [9]. The comparison is shown in Figure 3.9 and 3.10. The detection results obtained by the proposed method are given in Figure 3.9(a) & Figure 3.10(a). The proposed method is able to detect and differentiate various structures like facades, windows, and motifs. Further, it utilizes feature recovery to estimate the missing features (shown as square keypoints). Compared to this, the detection results obtained using Torii's reptile method in Figure 3.9(d) & 3.10(d) contain numerous clustering errors. As seen from the zoom-in area in Figure 3.9(c), the method mis-classify different types of repetitive patterns into a single class (label 1). In contrast, each type of elements like facades, windows are correctly clustered in our proposed method in Figure 3.9(b). Similarly, in Figure 3.10(b), different structures from walls and gates are correctly differentiated and clustered into different pattern types (labels 1, 2 & 5). However, Torii's reptile method grouped them into the same pattern type (label 1) in Figure 3.10(c).

The proposed method can also estimate occluded repeated features using Hough transform-based feature interpolation. An example can be seen in Figure 3.10(b),

where an occluded feature is shown in white diamond (label 5). It can be seen from the results that the proposed LS-RLF method is more robust and discriminative in estimating the lattice pattern and cluster structures. This will result in improvement in the subsequent image retrieval.

3.4.4 Performance of LS-RLF Visual Search

The performance of the proposed LS-RLF framework is compared with the Philbin's method [19], burstiness weighting (BRST-IDF) [8] and Torii's reptile method [9]. The mAPs obtained using these methods with 200K and 1M visual vocabularies are compared with those obtained by the proposed LS-RLF-based retrieval. The results are summarized in Table 3.1.

For the Oxford dataset, the proposed method with 1M visual vocabulary vocabulary (VV) achieves an mAP of 76% which outperforms the Philbin's method by 12.4%, Jegou's burstiness weighting by 7.2% and Torii's reptile method by 4.5%. This clearly shows the effectiveness of the proposed method. It is also observed that for all the methods, retrieval with 1M vocabulary consistently outperforms that with 200K. It is because larger vocabularies are more expressive compared to the smaller ones. The image representations with fine-grained visual words are more discriminative and hence achieve better search performance.

A similar trend of retrieval performance can be observed for the Paris dataset. The proposed LS-RLF framework with 1M VV achieves an mAP of 80.1% which outperforms the Philbin's method by 7.5%, Jegou's burstiness weighting by 6.9% and Torii's reptile method by 5.5%. Similarly, for the Holidays dataset, it can be observed that our proposed method outperforms other methods for both 200K and 1M VVs. The proposed method achieves an mAP of 75.9% and 84.1% for 200K VV and 1M VV, respectively. These results outperform the Torri's Reptile method by 3.2-3.9%, Jegou's BRST-IDF method by 5.4-7.8% and Philbin's method by 5.8-9.0%.

The designed weighting strategy is able to obtain balanced BoW histograms which allow unbiased similarity measure between images. The mAP obtained using the proposed method clearly show a superior performance when compared with other methods.

Figure 3.11 shows the qualitative retrieval results using the proposed LS-RLF framework. The query images with the search bounding box are given on the left column, while the top 10 retrieved images are shown in the corresponding rows in

Figure 3.9: Example 1: Repetitive pattern detection. The image on the top row (a) is obtained using the proposed LS-RLF method, the bottom row (d) is based on the method in [9]. Different groups of repetitive patterns are shown with different colors and annotated labels. Undetected but recovered new features are shown as square keypoints. See Section 3.4.3 for more explanations.

Figure 3.10: Example 2: Repetitive pattern detection on an example image. The image on the top row (a) is obtained using the proposed LS-RLF method, the bottom row (d) is based on the method in [9]. See Section 3.4.3 for more explanations.

Table 3.1: Performance comparison of the proposed LS-RLF framework on Oxford, Paris, and Holidays dataset.

Method	Oxford Dataset		Paris Dataset		INRIA Holidays	
	200K VV	1M VV	200K VV	1M VV	200K VV	1M VV
Philbin's method [19]	57.74	63.65	70.44	72.56	66.9	78.3
BRST-IDF [8]	63.09	68.86	72.14	73.24	68.1	78.7
Repttile Method [9]	67.72	71.60	74.10	74.64	72.0	80.9
Proposed LS-RLF Framework	70.25	76.06	78.63	80.11	75.9	84.1

Figure 3.11: Qualitative retrieval results using the proposed LS-RLF framework. Left-most column shows query images with the bounding box and the corresponding rows show the top 10 retrieved images.

the right. Images with green outlines are true positives (with respect to the ground truth), whereas those with red borders are false-positives. From the figures, it is clear that the proposed method can offer good retrieval performances.

3.4.5 Ablation Studies: Effect of Weighting Factors

In this section, we study the impact of the three factors (repetitive score (w_R), saliency weighting (w_S), and soft-assignment strategy (w_{SA}) on the retrieval performance of the LS-RLF framework. As w_{SA} is an integral part of the proposed method, it is included in all the case studies below. We will evaluate the importance of each weight by successively adding the repetitiveness score (w_R), and saliency weighting (w_S) into the proposed method. Table 3.2 shows the impact of different factors on the retrieval performance.

From Table 3.2, it can be seen that when each weighting factor is included into the proposed method, the mAP of the system improves for all datasets. For example, in Case 1 with the 1M visual vocabulary, when only the soft-assignment weighting is used, the method achieves an mAP of 66.38% and 73.29% for the Oxford and Paris dataset respectively. For Case 2, with the 1M visual vocabulary, when the soft-assignment weighting is used together with the repetitiveness score, the method offers a further mAP improvement of around 8.7% and 4.2% for the Oxford and Paris datasets respectively. This demonstrates the advantage of performing

Table 3.2: Ablation Study: Impact of Different Weighing Factors on Retrieval Performance of LS-RLF framework

Cases	Weights	Oxford Dataset			Paris Dataset			INRIA Holidays		
		200K VV	1M VV	1M VV	200K VV	1M VV	1M VV	200K VV	1M VV	1M VV
Case 1	(w_{SA})	60.38	66.38	73.29	72.00	73.29	72.38	81.60		
Case 2	$(w_{SA} + w_R)$	68.30	75.12	77.52	77.33	77.52	74.18	82.92		
Case 3	$(w_{SA} + w_R + w_S)$	70.25	76.06	80.11	78.63	80.11	75.91	84.10		

repetitive pattern detection. For Case 3, with the 1M visual vocabulary, when all the soft-assignment weighting, repetitiveness score, and saliency weighting are used, the proposed method offers a further mAP improvement of 1-3%. This demonstrates the advantage of integrating saliency information into the proposed method. Similar trends are observed for the Holidays dataset. These results show that each weighting has an impact on the performance, and the repetitiveness score appears to have a stronger impact on the performance than the saliency weighting.

3.4.6 Performance of FRS-based Visual Search

In order to demonstrate the effectiveness of the FRS metric, we perform experiments using visual vocabulary of 1M visual words. We first present case studies shown in Figure 3.12. The figure shows retrieval performances for sample query images using the proposed FRS metric and the Repttile method [9]. The top row shows an Oxford query and its precision-recall curve where our FRS method covers more area under the curve than the Repttile method. For this query, the Repttile method achieves an AP of 70.6%. Using the unique features alone achieves an AP of 73%. The FRS metric uses information of repetitive patterns and complements it with unique features to increase the AP to 81.2%. Similar trends can be observed for the image in the bottom row in the figure, which shows a query image from the Paris dataset. Hence, this shows that the proposed method utilizes repetitive patterns to enhance the retrieval performance.

Table 3.3: Performance comparison of the proposed FRS metric on Oxford, Paris, and Holidays dataset.

Method	Oxford Dataset	Paris Dataset	Inria Holidays
Philbin's method [19]	63.7	72.6	78.3
Burstiness Weighting [8]	68.9	73.2	78.7
Repttile Method [9]	71.6	74.6	80.9
Neural Codes [30]	54.5	-	78.9
Local CNN features [34]	64.9	69.4	83.6
NetVLAD [143]	70.8	78.3	82.1
Proposed FRS w/o QE	73.1	77.2	85.6
Proposed FRS	77.3	81.1	86.3

Table 3.3 presents mAPs obtained for the proposed FRS framework and comparison with other state-of-the-art methods. For the Oxford dataset, the proposed FRS framework achieves an mAP of 77% which outperforms Philbin's method [19]

Figure 3.12: Example queries images from the Oxford (top row) and Paris (bottom row) datasets with Average Precision

method by 13.6%, burstiness weighting [8] by 8.4% and Torii’s Reptile method [9] by 5.7%. Similarly, for the Paris dataset, the proposed method achieves an mAP of 81% with surpasses Philbin’s method by 8% [19], the burstiness weighting [8] by 7.9% and Torii’s Reptile method [9] by 6.5%. A similar trend can be observed for the Holidays dataset.

We also compare our method with deep learning based approaches [30, 34, 143], where global features extracted from CNN networks are used for image retrieval. As shown in Table 3.3, our method outperforms deep learning methods. This demonstrates the importance of repetitive pattern detection for image matching, which is not considered in current deep learning based approaches [30, 34, 143].

Overall, the existing state-of-the-art methods either do not consider or down-weight/limit the contribution of repetitive features which results in visual information loss. In contrast, the proposed FRS framework detects the repetitive patterns and offers an effective representation via proposed FRS metric and hence achieves better retrieval performance. This clearly demonstrates the effectiveness of the

proposed FRS-metric in visual landmark search.

3.5 Summary

In this chapter, we studied handcrafted feature-based visual search techniques. Specifically, we studied the repetitive patterns in images and the burstiness issue in visual search. We first proposed a new LS-RLF method which integrates local features and underlying lattice structure to detect repetitive patterns. A new feature interpolation technique is also presented to estimate and recover undetected or missing features. The key advantage of the LS-RLF framework is that the detected repetitive features can be used directly in conjunction with existing visual search technologies. In addition, the proposed method is able to alleviate the burstiness issue in BoW-based image search.

In order to exploit the information in repetitive pattern while handling the burstiness issue, we also proposed a new feature repetitiveness similarity (FRS) metric. The proposed method makes effective use of the repetitive patterns in the images to improve the image matching and search performance. Based on the detection, the repetitive features and unique features are handled separately, which help in addressing the burstiness issue in visual search. We demonstrated that the repetitive patterns are not a nuisance but can be used to complement the unique features in visual recognition. Experimental results on three benchmark landmark datasets clearly show the effectiveness of the proposed methods.

Chapter 4

Visual Fashion Search

4.1 Introduction

In this chapter, we will develop deep learning methods for visual fashion search. Visual fashion search is a fascinating application in computer vision which has received a huge research attention [10, 11, 41, 144, 145] and has been successfully deployed in large e-commerce companies [26, 27, 146] such as *eBay*, *Amazon*, *Pinterest*, *Flipkart*, etc. Learning discriminative and robust image representations and measuring visual image similarity is the key task in visual fashion search. This chapter will study various deep feature extraction and deep metric learning methods for clothing image representation. We will develop visual fashion search frameworks which consider the essential elements of fashion such as category, attribute, and brand.

Fashion demonstrates diverse visual concepts with their designs, styles, attributes, and brands. There exist a hierarchy of similarities between fashion clothing, ranging from exact instance or brand to similar attributes and styles. An effective fashion search method, thus, should be able to represent the different tiers of similarities and capture various fashion concepts. However, there exist several challenges to visual fashion search such as potential deformations in clothing shape, pose variations with presence/absence of models/mannequins in a diverse background. Moreover, brand information such as logos in clothing images are quite small and often suffer from various deformations. All these factors pose significant challenges to visual fashion search.

Clothing retrieval is an emerging application and has been studied in various

literatures. The earlier works [147–150] used handcrafted features to describe the clothing attributes and perform the fashion search. In recent years, deep learning [65] methods for fashion search have been developed in [10, 11, 32, 116, 151, 152]. Features from fined-tuned CNN networks have been used for clothing recognition and retrieval in [32, 151, 152]. The work in [32] used features extracted from CNN network fined tuned on clothing attribute database. Similarly, [151] used a latent layer after fully-connected (FC) layer to obtain hash-like representations from a CNN network for efficient retrieval. Similarly, [146, 152] also used FC layer features for fashion image representation and clothing retrieval. However, these methods directly use features extracted from CNN layers (FC layer features or CNN activations) but they do not perform similarity learning which is desirable for retrieval problems. Hence, such features are less discriminative for fashion retrieval and search.

In order to learn discriminative features using deep learning, deep metric learning methods [37, 40, 111, 112] have been employed for visual fashion search. Where-To-Buy-It (WTBI) method [10] used pre-trained CNN features and learned a similarity network using a siamese network. The multitask learning with attribute information has been used in [129, 144] for fashion representation. Dual Attribute-Aware Ranking Network (DARN) [129] used a siamese network with attribute information for the exact clothing search. Recently, Liu *et al.* proposed FashionNet [11] which used a triplet network to learn feature embedding using a multitask network for joint category and attribute recognition.

Nevertheless, currently available fashion datasets [11, 153, 154] and fashion search methods [10, 11, 26, 151, 152] have two major drawbacks. First, the existing datasets have annotation only for clothing category and attributes but do not explicitly consider brand information which is an essential element of fashion. Customers are often brand conscious and hence the brand information plays an important role in fashion search. Brand recall is a desirable trait in clothing retrieval. Second, existing methods do not represent different granularities of fashion clothing similarity. The feature space should ideally capture the hierarchy of similarities between fashion clothing for effective fashion search.

In view of this, this chapter first presents a new clothing dataset called NTUBrand-Fashion (NBF) dataset with 10K fashion images which are richly annotated with clothing categories, attributes, and brand information to capture the essential elements of fashion. The main advantages of this dataset over the existing ones are: (i) it has brand-level annotation which can be leverage to improve the per-

formance of fashion search, (ii) it has 3-tiers of annotations (category, attribute, and brand) which allows to represent fashion images at different granularities. We demonstrate the importance of brand information with a new Brand-Aware Fashion Search (BAFS) framework. Further, in order to capture different granularities of similarities, we propose a new Attribute Supervised Metric Learning (ASML) with a multitask CNN network that jointly learns fashion attributes and image embeddings while taking category and brand information into account.

The rest of the chapter is organized as follows. Section 4.2 presents the NTUBrand-Fashion dataset in detail including its collection and available annotations. In Section 4.3, we present the brand-aware fashion search (BAFS) framework and demonstrate the importance of brand information. Section 4.4 presents the proposed attribute metric learning (ASML) framework for fashion search. The experimental results and discussion for the proposed BAFS and ASML frameworks are presented in Section 4.5. Finally, Section 4.6 provides a summary of the chapter.

4.2 NTUBrandFashion (NBF) Dataset

Fashion clothing demonstrate diverse visual concepts with their designs, styles and brands. In practice, clothing items are generally characterized by their category (e.g. dress, t-shirt), attribute information (e.g. dotted-print, striped), and brand information (e.g. Adidas, Puma). The publicly available fashion datasets such as DeepFashion [11], Semantic-Clothing-Attributes (SCA) dataset [154], Apparel-Style (AS) dataset [153] only contain annotations for clothing categories and attributes. However, they lack annotation on brand information which is one of the essential elements in fashion. Thus, we construct a new dataset, NTUBrandFashion, with rich annotations which incorporates category, attribute as well as brand information of the fashion items. Such brand level information can be leveraged to improve the performance of visual fashion search.

The NTUBrandFashion (NBF) dataset consists of about 10K clothing images from 15 popular brands with distinct logos. Compared to the existing fashion datasets [11, 153, 154], our NBF dataset contains clothing brand information annotated in the form of brand logos. We will show later that the brand information enables us to perform brand-aware fashion search which can significantly improve the clothing retrieval performance (See Section 4.3).

Figure 4.1 shows some sample images of the NBF dataset. The dataset contains

rich annotations to capture the essential elements of fashion. Figure 4.2 shows additional example images from the dataset. Note that the clothing images have diverse backgrounds, various deformations and occlusions, and presence/absence of human models or mannequins. Also, brand logos occur at various scales and locations and often only cover a small region. These pose significant challenges to brand-aware fashion search.

Figure 4.1: Sample images from NBF dataset with annotations.

4.2.1 Image Collection

We follow the standard practice of dataset construction similar to DeepFashion dataset [11]. We crawled the images from Google Images¹ using relevant search keywords. Although we use Google search, further works of dataset annotations and cleaning are done manually by humans. For each brand, we first created a pool of keywords such as *Denim Versace Hoodie*, *Playboy Tee*, *Floral Adidas Tank* etc., to query the website. For 15 brands, around 150K images were downloaded. Irrelevant images were then removed by human screening. We also removed single channel and low-resolution images which are less than 300×300 pixels.

Next, to remove redundant and highly similar images, we used the FC7 features from VGG16 [24]. For each brand, pairwise image similarities were computed using FC7 features. For those images with similarity scores greater a threshold ($thres = 0.97$), we kept one copy and removed the rest. At the end of this step, the dataset contained 9,498 clean and relevant images.

4.2.2 Dataset Annotations

The NBF dataset is richly annotated with three tiers of information, namely, clothing category, attribute, and brand logo which captures essential elements of fashion. We develop annotation software to effectively annotate the images in the dataset. Figure 4.4 (a) shows a sample screenshot of the software tool developed for dataset annotation.

¹<https://images.google.com>

4.2. NTUBRANDFASHION (NBF) DATASET

Figure 4.2: Sample images from the NBF dataset.

A. Clothing Category and Bounding Box Annotation

Fashion images in the NBF dataset are categorized into 16 clothing classes. Each image is first labeled with clothing category together with their bounding box coordinates. The category labels are useful for the clothing classification task. The bounding box annotation can be leveraged for clothing detection to reduce the impact of background clutter during the visual search. Table 4.1 lists all the 16 clothing categories in the NBF dataset.

Table 4.1: List of 16 clothing categories in the NBF dataset

Upper-Body	Lower Body	Full-Body
Button-Down	Capris	Dress
Cardigan	Joggers	Robe
Hoodie	Leggings	Romper
Jacket	Shorts	
Sweater	Skirt	
Tank	Sweatpants	
Tee/T-shirt		

B. Brand Logo and Bounding Box Annotation

Similar to the clothing categories, all the brand logos in images are annotated with corresponding brand labels and bounding box coordinates. These annotations are used for logo detection to explore the clothing brand information. This dataset incorporates 15 popular fashion brands such as *Adidas*, *Nike*, *Puma* etc. Sample graphic logos from each of these brand are shown in Figure 4.3. Figure 4.4 shows the screenshot of the tool developed for dataset annotation together with example images where fashion images are annotated with clothing categories and brand logos.

C. Clothing attribute annotation

In order to describe the image in finer detail, we annotated images with 32 at-

Figure 4.3: Sample logos of 15 brands in NTUBrandFashion dataset. (a) Adidas, (b) All Star Converse, (c) Abercrombie and Fitch (d) Ferrari, (e) Kappa, (f) Levi's, (g) New Balance, (h) Nike, (i) The North Face, (j) Playboy, (k) Puma, (l) Ralph Lauren, (m) Reebok, (n) Vans, and (o) Versace

Figure 4.4: (a) Screenshot of the developed annotation tool. (b) Sample fashion images annotated with clothing categories and brand logos with corresponding bounding boxes.

tributes based on visual concepts such as color, part, pattern, etc. These attributes are commonly used to describe the clothing by the human in practical scenarios (for example *black-dotted dress*). Such attributes can be used to train and can enhance the retrieval performance. The list of the semantic attributes is presented in Table 4.2. Figure 4.5 shows example images representing attributes related to clothing patterns. Such information can provide rich information about fashion

4.3. BRAND-AWARE FASHION SEARCH (BAFS)

Table 4.2: List of semantic attributes in NBF dataset

Color	Pattern	Person	Sleeve-type	Neck-type	Parts
Black	Animal print	Male	Long	V-neck	Pocket
Blue	Diamond	Female	Short	Round/Crew	Hooded
Green	Checkered		Sleeveless	Collared	Zipper
Brown	Dotted				Button
Grey	Floral				Crop
Orange	Striped				
Pink	Color-block				
Purple	Graphics				
Red					
White					
Yellow					

clothing.

Figure 4.5: Sample fashion images representing various types of 'pattern' attributes.

4.3 Brand-Aware Fashion Search (BAFS)

In this section, we will present a new brand-aware fashion search (BAFS) framework which explores the clothing brand information during fashion search. We study the importance of brand information for fashion search and demonstrate the usefulness of our NBF dataset.

Consider a case where a user would like to search a branded clothing item. Fig. 4.6 shows examples of clothing retrieval using CNN features extracted from VGG16-Net [24]. The images in the first column show the query images and corresponding retrieved images are shown in the respective rows. For instance, in the first row, the customer wants to search for a black *Adidas* hoodie. Although the search has retrieved visually similar images, the results are not brand aware. In this scenario, the hoodies from other brands such as *Puma*, *Ferrari* etc. are also

Figure 4.6: Example of branded clothing search. Relevant and irrelevant retrieved images are shown with green border and red border respectively

retrieved in the top ranks. Hence, the search method only focuses on high-level global representation but lacks consideration of small logos.

In order to alleviate such issue, we propose a brand-aware fashion search framework. The overview of the proposed framework is shown in Fig. 4.7. Given a query image of a fashion item, the framework first detects the clothing region and brand logo in the image. Next, using the detected clothing region, we extract discriminative feature using the activations of CNN across different layers. We compute the similarity score between the query and database images to rank the database images. The initial retrieved images are then passed through a brand-aware re-ranking engine and the framework returns the relevant clothing images to the user. The three main modules in the proposed method are described in the following sections.

4.3.1 Joint Clothing and Logo Detection

The proposed framework simultaneously detects the existence of clothing items and brand logo in images. We develop a joint clothing and logo detector using Faster-RCNN [22]. The detector contains two sub-networks namely, Region Proposal Network (RPN) and Fast-RCNN which shares common convolutional layers for efficiency. In order to generate the region proposals for clothing and logos, RPN slides the pre-defined anchor boxes of different scales and aspect ratios over the last convolution layer. The original work [22] uses anchor boxes from 3 different

Figure 4.7: Proposed Brand-Aware Fashion Search (BAFS) framework

scales. In this work, in order to target small brand logos, we sample the anchors from finer scales and use anchor boxes from 4 different scales. The RPN is trained to regress the bounding boxes from anchor boxes using a multi-loss function in (4.1).

$$L(p_i, b_i) = L_{cls}(p_i, p_i^*) + \lambda \cdot p_i^* L_{reg}(b_i, b_i^*) \quad (4.1)$$

The first term in (4.1) refers to soft-max loss used to classify the anchors as objects or non-objects. p_i and p_i^* are the predicted probability for anchor i being an object and the ground-truth label respectively. The second term is activated only for true anchors ($p_i^* = 1$) and it represents the regression loss for bounding box prediction, where b_i and b_i^* are predicted and ground-truth box coordinates.

Next, the obtained bounding boxes are used to pool the features from the last convolutional layer and the features are used to classify clothing items and brand logos. We use a 4-step alternation method [22] to train the detector network.

4.3.2 PMAC Feature Extraction

Several previous works have exploited the fully-connected layer features [27, 30–32, 146, 151, 155] and pooled features from convolutional layers [35, 156] for image retrieval. However, these features do not encode the hierarchy of information required for fashion search. Thus, we propose a new Principal Component Maximum Activation of Convolutions (PMAC) feature encoding which leverages both low-level and high-level abstractions to extract rich features for retrieval. Typically, the early layers of CNN encode low-level features such as lines, edges, colors etc., the mid-layers encode part-based features such as patterns, logos etc., while

the late layers provide a high-level inference about the class or overall looks of the image [72]. Our encoding method tries to capture all these features into a single descriptor.

In order to extract the PMAC features from images, we first operate on the n activations of convolutional layers $L = \{l_i\}_{i=1}^n$ of the network. For a particular layer l , we pool feature from its 3D activation maps of $W^{(l)} \times H^{(l)} \times K^{(l)}$ dimensions, where $K^{(l)}$ is the number of filters in layer l and $W^{(l)} \times H^{(l)}$ represents the spatial region $R_i^{(l)}$, $i = \{1, 2, \dots, K\}$ of the feature maps. For each feature map $R_i^{(l)}$, we spatially pool the maximum activation of the map to construct a feature $\mathbf{f}^l = \{f_1^l, f_2^l, \dots, f_i^l, \dots, f_K^l\}$ for layer l .

$$f_i^{(l)} = \max(R_i^{(l)}), \text{ where, } i = 1, \dots, K \quad (4.2)$$

Next, we concatenate the features from multiple layers to generate a single descriptor $\mathbf{F} = [\mathbf{f}^l]$, $l \in \{1, \dots, n\}$. The concatenation strategy is a simple yet effective way for image representation. Next, in order to extract discriminative information, the features $\mathbf{F}$ are projected to a new subspace using Principal Component Analysis (PCA) and whitening. Mathematically, we construct k -dimensional PMAC feature vector $\mathbf{x} = \{x_1, x_2, \dots, x_i, \dots, x_k\}$. Each component x_i of the vector is computed using Eq.(4.3),

$$x_i = \frac{1}{\sqrt{\lambda_i}} \mathbf{U}^T \cdot F_i \quad (4.3)$$

where $\mathbf{U}$ represents the transformation matrix formed by k -principal eigenvectors and λ_i represents the corresponding eigenvalues. Using this feature representation, the similarity between a query image $\mathbf{q}$ and database image $\mathbf{d}$ is computed using the cosine similarity measure given by,

$$\text{sim}(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q}^T \mathbf{d}}{\|\mathbf{q}\|_2 \|\mathbf{d}\|_2}. \quad (4.4)$$

4.3.3 Brand-Aware Re-ranking

Although the PMAC image encoding captures both the low-level details and high-level abstraction of the images, it does not prioritize the brand information. This is because brand logo information generally tends to cover a small area in images, thus visual information from such small objects is not captured well into the

descriptor. The proposed framework thus uses a brand-aware re-ranking engine to rank the initial retrieval list. To achieve this, we use information from the detected logo using the joint logo and clothing detector. From the list, we first retrieve the clothing images whose brand match with the detected logo. The retrieved images are then ordered at the higher ranks, while other images are pushed to the lower ranks in the list. Hence, we obtain a re-ranked list of images based on brand. We also perform similar re-ranking based on the detected clothing category. The top-ranked images are returned by the framework as the retrieved images.

We will demonstrate the performance of PMAC feature and brand-aware re-ranking engine in Section 4.5.4.

4.4 Fashion Search using Attribute-Supervised Metric Learning

In the previous section, we presented a visual fashion search framework based on the features extracted from CNN layers. Such features may be less discriminative for fashion search as the method do not perform similarity learning which is desirable for retrieval problems. In this section, we present a new deep metric learning method called *Attribute Supervised Metric Learning* (ASML) for learning discriminative and robust image representations for visual fashion search.

Our aim is to learn fashion similarities at different granularities demonstrated by diverse visual concepts in fashion. We present an end-to-end approach to learn such similarity space where the closeness between the images embeddings corresponds to their visual similarities. For example, Fig. 4.8 illustrates the desired feature space with clothing images, where the distances between the center reference image (R) and images (A)-(E) are represented by the dotted arrows. Image (A) is closest to the reference image (R) and lies inside the inner circle since image (R) and image (A) are the same instances of a particular clothing item. Image (B) is similar to image (R) with variation in color and is second nearest to it. Similarly, images (C) and (D) are of the same category with different pattern and design and hence lie farther apart. Finally, the out-of-category image (E) lies farthest from image (R). Hence, image similarities occur at different granularities. We aim to learn such a feature space using deep metric learning with supervision at different granularities of image similarities.

Triplet networks and triplet loss have been shown to achieve state-of-the-art per-

Figure 4.8: Illustration of the desired feature space for the visual search and recommendation. Example clothing images exhibit similarities at different granularities represented by the concentric circles. The distances between the reference image (R) and images (A)-(E) reflect different degrees of visual similarity, with respect to image (R); visually similar images are closer while dissimilar images lie farther apart.

performances in metric learning [37, 123]. However, the existing methods based on triplet networks [37, 115, 123] focus on discovering instance level similarity and do not effectively represent similarity at multiple granularities as shown in Figure 4.8. The triplet selection strategies [37, 123, 157] employ hard binary decisions during the selection process. In fact, the training triplet samples have different degrees of information and capability to teach the network. As the existing methods treat all the triplets with equal importance, this prevents the network to learn different levels of similarity.

In view of this, we propose an ASML framework that uses multitask learning and considers the interactions between the attribute prediction and triplet network by jointly training them. This has two major advantages over the existing methods. First, it helps in mining informative triplets especially when exact anchor-positive pair annotations are not available. Second, training samples are treated based on their importance in a soft manner which helps in capturing different tiers of similarity required for fashion search. The following sections describe our proposed ASML framework in detail.

Figure 4.9: Architecture of the proposed attribute-supervised metric learning (ASML) network.

4.4.1 Overview of the ASML Framework

Our goal is to compute an embedding $f_{\theta}(x)$ for image x , which can be used for the image representation. The distance between the images in the embedding space should reflect their visual similarity in the image space, thus capturing different levels of similarity as shown in Figure 4.8.

In order to learn such an image embedding space, we propose a new deep metric learning method using attribute-supervised triplet network. The overall architecture of the proposed framework is shown in Figure 4.9. The network consists of three identical convolutional neural networks (CNNs) with shared parameters θ , which are to be learned using the proposed attribute supervised triplet training. The network takes three input images, a reference anchor image (x^a), a positive image (x^p) which is from the same class/ID as the anchor, and a negative image (x^n) which is from a different class/ID.

The last fully connected layer in each network has two branches for multitask learning, which (i) computes the feature embedding $f(x)$, and (ii) produces the attribute vector $\mathbf{v}$ that predicts the attributes in the image. The most important part of the proposed framework is that the two tasks not only are simultaneously

optimized, but also linked by a supervising signal which incorporates attribute information. The supervising signal assists in the triplet sampling and training for learning differences between the images. The attribute-supervised triplet network is optimized by a combined attribute and supervised triplet loss given by

$$L_{total}(\theta) = L_{tri}^S(\theta) + \lambda L_{attr}(\theta) \quad (4.5)$$

where $L_{tri}^S(\theta)$ and $L_{attr}(\theta)$ represent the proposed supervised triplet loss and attribute loss respectively. The parameter λ is used to balance the contributions of the two losses. We will explain the significance of the attribute and supervised triplet loss in the following sections.

4.4.2 Attribute Prediction

Attribute information has been used to enhance the performance in various problems such as classification [158, 159], food recipe retrieval [160], person Re-ID [127, 128], and fashion search [11, 129]. The attributes provide useful information about the images. For example, *spotted-skin*, *claws*, *tail-length*, etc. are used to describe animals [158], and *color*, *texture*, *sleeve-length*, etc. of clothing are used to describe fashion images [11, 129]. In this work, we explore and incorporate these attribute information to enhance the triplet network training.

A set of K semantic attributes are used to describe the image appearance. Let us denote the image attributes by $\mathbf{a} = [a_1, a_2, \dots, a_K]$, where each element $a_i \in \{0, 1\}$ denotes the presence or absence of the i^{th} attribute in a particular image. We treat the problem of attribute prediction in an image as multilabel classification where an image can have multiple attributes. To achieve the attribute prediction, we pass the first branch from last fully connected layers into a sigmoid activation to squeeze the output to within $[0, 1]$, which produces the output $\mathbf{v}$. The attribute prediction loss $L_{attr}(\theta)$ is defined using binary-cross entropy loss function as follows.

$$L_{attr}(\theta) = - \sum_{i=1}^K [w_{pos} a_i \log(v_i) + w_{neg} (1 - a_i) \log(1 - v_i)], \quad (4.6)$$

where a_i is binary target attribute labels for image x , and v_i are components of $\mathbf{v} = [v_1, v_2, \dots, v_K]$, which is the predicted output attribute distribution. The weights w_{pos} and w_{neg} are used to balance positive and negative samples and determined by their average ratios in training labels. Note that the vector $\mathbf{v}$ is not binary, but is an attribute distribution, where each element v_i denotes the probability

of the existence of attribute a_i in image x , *i.e.* $p(a_i|x)$. For the triplet input $\{x^a, x^p, x^n\}$, $\{\mathbf{v}^a, \mathbf{v}^p, \mathbf{v}^n\}$ represent the attribute vectors which describe the images in the attribute space. We exploit this information to supervise and effectively train our proposed triplet network which is explained in detail in the following section.

4.4.3 Attribute Supervised Triplet Network

The performance of the triplet network depends mainly on two factors, namely the triplet selection and the triplet loss function. Both of these factors play significant roles in optimal learning and network convergence during the training process. In this work, we supervise both triplet selection as well as the loss using information from the attribute prediction of the multitask framework.

A. Triplet Mining

Random sampling based on class/ID labels for triplet does not assure the selection of informative examples. In cases where only category level information is available, the selected triplet examples may contain limited information for training, as the anchor-positive pairs from the same class can demonstrate a diverse visual appearance (*e.g.* clothing images of a dress with different colors, cloth materials, patterns may look very different). Thus, we first select the triplets with reliable anchor-positive pairs. The work in [26] and [40] used additional features and manual verification for the selection of anchor-positive pairs in the triplets, which is very expensive and prohibitive for large datasets. Instead, we propose to leverage cosine similarity, $\langle \mathbf{p}, \mathbf{q} \rangle = \frac{\mathbf{p}^T \mathbf{q}}{\|\mathbf{p}\|_2 \|\mathbf{q}\|_2}$, between the anchor-positive attribute vectors to sample and obtain reliable triplets. We use a threshold (Φ) such that triplets with $\langle \mathbf{v}^a, \mathbf{v}^p \rangle > \Phi$ are selected for the triplet training. This ensures that the anchor-positive pairs in the triplets are similar in attribute space and hence are reliable.

B. Attribute Supervised Triplet Training

Next, in order to exploit different degrees of triplet importance during the training, the triplet loss is supervised using the information from the proposed multitask architecture. Let $\{x^a, x^p, x^n\}$ be an input triplet sample to the proposed network, which produces the embeddings $\{f(x^a), f(x^p), f(x^n)\}$ and three attribute vectors $\{\mathbf{v}^a, \mathbf{v}^p, \mathbf{v}^n\}$. The conventional triplet loss over this triplet is given by Eq. (4.7).

$$L_{tri}(\theta) = \left[\|f_\theta(x^a) - f_\theta(x^p)\|_2^2 - \|f_\theta(x^a) - f_\theta(x^n)\|_2^2 + m \right]_+, \quad (4.7)$$

where $[z]_+ = \max(0, z)$ is the hinge loss, and θ represents network parameters. The direct use of this loss function does not consider the information from the image attribute prediction. We formulate our attribute supervised triplet loss training based on the multitask outputs as follows.

$$L_{tri}^S(\theta) = w(a, p, n) \left[\|f(x^a) - f(x^p)\|_2^2 - \|f(x^a) - f(x^n)\|_2^2 + m(a, p, n) \right]_+ \quad (4.8)$$

where $w(a, p, n)$ and $m(a, p, n)$ are the loss weighting factor and margin factor, which are determined by the output attribute distributions of the anchor, positive and negative images. Our motivation is that the closeness of images in attribute space can be estimated by comparing their attribute distributions. Hence, by comparing similarities between the attribute distributions of the anchor-positive pair and anchor-negative pair of a triplet, we estimate information contained in the triplet, which in turn, is used to supervise the triplet training. Following this, we propose two strategies to tune the loss with the supervising signal using the functions $w(a, p, n)$ and $m(a, p, n)$.

B1. Soft-Weighted (SW) Triplet Loss (ASML-SW)

The soft-weighted (SW) triplet loss $L_{tri}^{SW}(\theta)$ operates on the overall loss using $w(a, p, n)$ while setting $m(a, p, n)$ to a constant margin m_0 , and is given by

$$L_{tri}^{SW}(\theta) = w(a, p, n) \left[\|f(x^a) - f(x^p)\|_2^2 - \|f(x^a) - f(x^n)\|_2^2 + m_0 \right]_+. \quad (4.9)$$

We propose to use $w(a, p, n) = \langle \mathbf{v}^a, \mathbf{v}^p \rangle \langle \mathbf{v}^a, \mathbf{v}^n \rangle$, the product of similarities between attribute vectors of anchor-positive and anchor-negative pairs. The above function adaptively alters the magnitude of the triplet loss. When the anchor-positive pair is similar in attribute space (*i.e.* $\langle \mathbf{v}^a, \mathbf{v}^p \rangle$ is high), the triplet sample is more confident and reliable. On the other hand, when the anchor-negative pair is similar in attribute space (*i.e.* $\langle \mathbf{v}^a, \mathbf{v}^n \rangle$ is high), it forms a hard negative example and carries high information. For example, a black tee and a black women's top have similar attributes such as short sleeve, crew neck, black color, etc. This means although the pairs are from different class/ID, but look very similar in attribute space, then the triplet should be given higher priority and more attention during the network update. Thus, when both $\langle \mathbf{v}^a, \mathbf{v}^p \rangle$ and $\langle \mathbf{v}^a, \mathbf{v}^n \rangle$ are high, the triplet is more informative and hence the network pays more attention to learn the differences. This is analogous to hard negative mining [37], but we handle them in a soft-manner.

Thus, the proposed loss provides a way to incorporate the importance of triplets and learns similarity at multiple levels.

B2. Soft-Margin (SM) Triplet Loss (ASML-SM)

The soft-margin triplet loss operates on the margin parameter using $m(a, p, n)$ while setting $w = 1$. Naive triplet loss uses a constant margin (m), which treats all triplets equally and restricts learning multilevel similarity as shown in Figure 4.8. We propose to use soft-margin where the term $m(a, p, n)$ in Eq.(4.8) is determined by attribute information and provides the effective margin,

$$m(a, p, n) = m_0 \log(1 + \langle \mathbf{v}^a, \mathbf{v}^p \rangle \langle \mathbf{v}^a, \mathbf{v}^n \rangle) \quad (4.10)$$

Similar to soft-weighted triplet loss, when both $\langle \mathbf{v}^a, \mathbf{v}^p \rangle$ and $\langle \mathbf{v}^a, \mathbf{v}^n \rangle$ are high, the triplet is more reliable and informative (hard negative). In other words, when the negative image and anchor image are very similar in the attribute space, a reliable margin is used to learn the difference and avoid the confusion. On the other hand, both $\langle \mathbf{v}^a, \mathbf{v}^p \rangle$ and $\langle \mathbf{v}^a, \mathbf{v}^n \rangle$ are low, the triplet is less informative. Here the anchor-positive pair is less reliable, and the anchor-negative pair forms an easy negative. Thus, the triplet generates less penalty if selected for training. The soft-margin adaptively handles this and incorporates it into the triplet loss which treats triplet samples based on their importance while prioritizing the informative ones. The attribute-supervised triplet loss training is summarized in Algorithm 3. The designed soft triplet loss can learn different levels of similarity required for visual search and ranking.

The learned feature embeddings from the network are used as image representations which are subsequently used in comparing image similarities. Image similarity is computed using the $L2$ -distance between the embeddings. The database images are ranked in a descending order of the computed distances and top-ranked images are returned by the framework.

4.5 Experiments

In this section, we will report the experimental results for the proposed frameworks. We first present the results and discussion for the proposed BAFS framework and demonstrate the importance of brand information and usefulness of the constructed dataset for fashion search. Next, we will present experimental re-

Algorithm 3 Attribute-Supervised Metric Learning (ASML)

Input: Images $\{x_i\}$ with class/ID labels $\{C_i\}$ and corresponding attributes

Output: Trained network parameters $\{\theta\}$

- 1: Initialize the parameters θ of the proposed network
- 2: **for** $t < T$ **do**
- 3: Select a triplet $\{x^a, x^p, x^n\}$ s.t. $\{x^a, x^p\} \in C_i$ and $x^n \notin C_i$
- 4: Compute outputs: $\{f(x^a), f(x^p), f(x^n)\}$ $\triangleright$ Embeddings
 $\& \{\mathbf{v}^a, \mathbf{v}^p, \mathbf{v}^n\}$ $\triangleright$ Attribute Vectors
- 5: Compute similarities $\langle \mathbf{v}^a, \mathbf{v}^p \rangle$ and $\langle \mathbf{v}^a, \mathbf{v}^n \rangle$
- 6: **if** ($\langle \mathbf{v}^a, \mathbf{v}^p \rangle \geq \Phi$) **then** $\triangleright$ Sampling
- 7: Compute $L_{attr}(\theta) = (L_{attr}^a + L_{attr}^p + L_{attr}^n)/3$ using Eq.(5.1)
- 8: Compute the supervising signals $w(a, p, n)$ and $m(a, p, n)$
- 9: Compute supervised loss: $L_{tri}^S(\theta)$ using Eq.(4.8)
- 10: Compute total loss: $L_{total}(\theta) = L_{tri}^S(\theta) + \lambda L_{attr}(\theta)$
- 11: Update: $\theta \leftarrow \theta - lr \frac{\partial_\theta L_{total}}{\partial \theta}$ $\triangleright$ Backward Pass
- 12: **end if**
- 13: **end for**
- 14: **return** θ

sults for fashion search framework using the attribute supervised metric learning (ASML) framework.

4.5.1 Datasets

We demonstrate the effectiveness of the proposed methods on fashion search using two datasets, namely NTUBrandFashion (NBF) dataset and DeepFashion dataset.

NTUBrandFashion (NBF) Dataset: We will use the NBF dataset introduced in Section 4.2. This dataset contains annotations at different granularities which serve to better assess the performance of search as well as recommendation. There are 50 query images in the dataset which are used for the performance evaluation. Note that we do not have exact positive pair annotations for triplet training.

DeepFashion Dataset: We also report results for large-scale DeepFashion dataset [11] in order to demonstrate the effectiveness of the proposed ASML framework. The DeepFashion in-shop dataset [11] consists of 52,712 images of 11,735 clothing items. These images are annotated with clothing IDs rather than category labels, where each clothing ID refers to exact same instances of clothes with varying angles, poses and viewpoints. Hence, positive-pair annotation for triplet training is available. The images are also annotated with 463 attributes. The images are divided into 12,612 database images, 14,218 query images, and 25,882 training images. Here the task is to retrieve the exact clothing items which are the same

item as the query.

4.5.2 Experimental Settings

For the BAFS framework, we use ZF-Net [72] and train it for the joint clothing and logo detection. The ZF-Net is a lightweight model with fewer parameters when compared to other architectures such as Alexnet, VGGNet, and ResNet. The network is initialized using a model pretrained on Imagenet [6]. We train the network using SGD optimization with a momentum update of 0.9 and initial learning rate of 0.001.

For the ASML framework, we use VGG16 [24] as our base network in the proposed architecture, where convolutional layers are followed by two fully connected layers with ReLU activation and dropout of 0.5. The network is trained using the loss defined in Eq. (4.5), with SGD momentum update of 0.5 and initial learning rate of 0.001. We treat both triplet and attribute losses equally and set the value of λ to 1. We implemented our methods using PyTorch framework [161].

The value of margin m_0 is experimentally set to 0.5 and 0.2 for NBF dataset and DeepFashion dataset respectively. Experiments show that small deviations from these values do not affect the performance much. For the soft-margin triplet loss, the initial values of m_0 for the NBF dataset and DeepFashion dataset are set to 0.8 and 0.3. We set the threshold (ϕ) to 0.7 and observe that the performance is fairly stable for $\phi \in [0.5, 0.9]$. L_2 -normalized feature from the last fully connected layer is used as the feature vector, which is optimized by both multitasks.

During the triplet selection from the NBF dataset, items from the same category and brand are sampled for the anchor-positive pairs, and items from different categories or brands constitute the negative samples. For DeepFashion dataset, the triplets are randomly sampled such that the anchor-positive pairs are sampled from the same item (*i.e.* same ID), and negatives are from a different item (*i.e.* different ID). Since the NBF dataset has only category-level information, we augment triplets from DeepFashion in order include instance-level triplets while training on NBF dataset. For fair comparison, the same set of training triplets is used for all the experiments and comparative methods. Following the previous works [10, 11, 26, 146, 151], we used Faster-RCNN [22] detector to detect the clothing region and reduce the impact of background clutter.

4.5.3 Evaluation Metrics

We evaluate the performance of the NBF dataset for instance search and recommendation. Visual instance search refers to finding the exact same instance (item/product) from the database, while the latter refers to providing a ranked list of images based on their visual relevance. We use mean Average Precision (mAP) as the performance metric for the instance search. Following [129], we used Normalized Discounted Cumulative Gain (NDCG) (4.11) to measure the performance of recommendation. The metric reflects emphasis on more relevant items being ranked at higher positions.

$$NDCG@k = \frac{1}{Z} \sum_i^k \frac{2^{r(i)} - 1}{\log_2(1 + i)} \quad (4.11)$$

Here the relevance of the i^{th} ranked image is calculated based on its matching with the query image. It considers three levels of information, namely category, brand and attribute. The relevance score is the sum of these three factors *i.e.* $r(i) = r_i^{cat} + r_i^{brand} + r_i^{attr}$, where r_i^{cat} represents the category match $\in \{0, 1\}$, r_i^{brand} is the brand match $\in \{0, 1\}$. As the attributes are multilabel, the attribute match r_i^{attr} is computed by taking the ratio of the number of matched attributes to the total query attributes [129]. Overall, the relevance score summarizes the similarity between the images at multiple granularities of category, brand and attributes.

For DeepFashion dataset, in order to make the comparison consistent, we use the same evaluation metric used by [10, 11] *i.e.* top- k retrieval accuracy, which considers a hit if the exact same item as the query is found in the top- k ranked list.

4.5.4 Results and Discussion: Proposed BAFS Framework

In this section, we first present results obtained for the proposed BAFS framework. We report the result on the NBF dataset which has brand information to evaluate brand-aware fashion search. In the following, we first present results for joint clothing and logo detection. We then provide an analysis on the choice of CNN layers for PMAC feature encoding and present the retrieval results of the proposed BAFS framework. Finally, we compare the performance of the proposed BAFS method with other comparative methods [32, 35, 146, 151].

A. Joint Logo and Clothing Detection

The detector network uses fixed sized anchor boxes to generate the region proposals. Typically, anchors from 3 different scales $\{128 \times 128, 256 \times 256, 512 \times 512\}$ are used to obtain the proposals [22]. In this work, we aim to detect the small brand logos from clothing images. Thus, we train our network using finer anchor boxes with 4 different scales $\{64 \times 64, 128 \times 128, 256 \times 256, 512 \times 512\}$. Table 4.3 shows the detection results in terms of average precisions (APs) for various clothing and logo categories using 3-scales and 4-scales anchor boxes. With 3-scale anchor boxes, the logo detection obtains an mAP of 60.82%. The logo detection performance is significantly improved with finer 4-scale anchor boxes which achieves an mAP of 77.88%, an improvement by an mAP of 17%. The performance boost is especially noticeable for logos such as *New Balance*, *Abercrombie & Fitch* etc. which often occur in very small scales. The experiments clearly show that finer scales of region proposals improve logo (small object) detection while maintaining similar performance for larger clothing categories. Overall, our detector achieves mAPs of 70.95% and 77.88% for clothing items and brand logos respectively.

Figure 4.10 shows examples for joint detection of logo and clothing. From the figure, we can observe that the detector can simultaneously detect various logos and clothing items in the fashion images under different scales, deformations, and background clutter. The detected information on clothing and brand is used in visual search framework to improve the retrieval results.

B. Retrieval Performance of the Proposed BAFS Framework

Table 4.4 shows the retrieval performance using different convolutional layers and their combinations. It is observed that the early layers achieve poor performance as early features are too generic which only represent low-level visual information such as edge, color, blobs etc. [72]. We observe that features from the third layer to the last layer provides a relatively similar performance ($\approx 30\%$). Although all of these layers provide similar performance, they encode different level of information. Therefore, in order to exploit a rich hierarchy of feature information, we explore various ways to combine layers for feature representation. The combined use of the three penultimate layers of the network shows the best performance ($\approx 35\%$). Hence, we choose these layers for feature encoding in our subsequent experiments.

The overall retrieval performance of the proposed framework is shown in Table 4.5. It shows the performance comparison between various features used and different re-ranking steps. The initial retrieval using direct concatenated feature vector achieves an mAP of 34.8%. The retrieval using the proposed PMAC feature out-

Figure 4.10: Sample detection of clothing and logo in images from NBF dataset. Detected items are indicated with bounding boxes with their confidence scores. Note that the detector can jointly detect both brand logos and clothing items under various deformations, background clutter, scales and orientations.

Table 4.3: Average Precision (%) of logo and clothing detection.

Clothing	3-scales	4-scales	Logo	3-scales	4-scales
Button-Down	66.49	62.77	Abercrombie & Fitch	34.87	76.93
Capris	52.36	49.13	Adidas	72.32	80.40
Cardigan	62.93	63.30	All Star Converse	77.38	87.19
Dress	65.58	67.35	Ferrari	59.59	76.44
Hoodie	86.38	86.22	Kappa	50.31	76.66
Jacket	88.26	88.45	Levis	77.39	79.49
Joggers	55.45	55.26	New Balance	28.05	64.39
Leggings	70.63	70.68	Nike	45.12	69.72
Robe	71.32	72.37	Playboy	62.28	69.38
Romper	81.35	80.74	Puma	57.90	75.87
Shorts	79.01	77.90	Ralph Lauren	80.35	88.60
Skirt	67.88	68.03	Reebok	57.86	76.04
Sweater	66.39	65.83	The North Face	42.87	70.36
Sweatpants	53.77	57.03	Vans	84.71	86.28
Tank	81.01	80.56	Versace	81.41	90.56
Tee	89.52	89.64			
mAP	71.14	70.95	mAP	60.82	77.88

Table 4.4: Retrieval performance (mAP %) Vs. convolutional layers used for feature encoding

Single Layer	mAP	Multi-Layers	mAP
conv1	8.9	conv3 + conv4	33.2
conv2	20.6	conv3 + conv5	33.0
conv3	29.9	conv4 + conv5	32.2
conv4	29.2	conv3 + conv4 + conv5	34.8
conv5	30.0		

performs by achieving an mAP of 45.9% (a gain of 11%). Although the direct concatenated features try to capture multi-layer feature information, it contains significant redundant information. In contrast, the proposed PMAC extract discriminative features from various layers which capture low-to-high level visual information. The initial performance of the proposed PMAC is further improved by brand-aware re-ranking to 51.9% and 53.6% using clothing and logo re-ranking. The experimental results clearly show the advantage of PMAC feature encoding and brand-aware re-ranking.

C. Comparison of BAFS with other Methods

Table 4.6 shows the performance comparison of the proposed BAFS framework with baselines and other comparative methods [32, 35, 146, 151] which uses vari-

Table 4.5: Retrieval performances of the proposed BAFS framework

Feature Used	Initial Retrieval	Clothing Re-ranking	Logo Re-ranking
Direct Concatenated	34.85	39.90	45.93
PMAC	45.93	51.92	53.60

ous features extracted from CNN layers. From the table, it is observed that the baseline method achieves the lowest performance as it directly uses features from the pre-trained networks which are not well adapted to the clothing domain. The R-MAC [35] method uses features pooled from the last convolutional layer and achieves an mAP of 30.03%. The other methods [151] and [146] use features extracted from the fine-tuned network on domain dataset and achieves 33.17% and 37.11% respectively. All of these methods rely on the features extracted from the single layer of the network which often does not capture the full range of information which is crucial for clothing instance search. Moreover, they do not effectively incorporate the brand information during the search. Compared to this, the proposed BAFS method uses a rich hierarchy of CNN features and incorporate the brand information, and hence it achieves an mAP of 53.6% which clearly outperforms other comparative methods.

The qualitative analysis on the retrieval performance of the proposed method is shown in Fig. 4.11, which further demonstrates the effectiveness of the proposed BAFS framework. For example, the bottom query in Figure 4.11 is an image of color-block *Kappa* hoodie. The RMAC method [35] retrieves a color-block *Addidas* hoodie with a different color at Rank 1. Similarly, rapid-clothing [151] retrieves a different model of hoodie at Rank 2. Compared to this, our BAFS framework successfully retrieves the clothing images which are the same items as the query. This clearly demonstrates that the brand information is crucial in fashion and incorporating it into the search framework can significantly improve the fashion search performance.

4.5.5 Results and Discussion: Proposed ASML Framework

In this section, we present results for fashion search using the proposed ASML framework. We report the performances of soft-weighted ASML (ASML-SW) and soft-margin ASML (ASML-SM) on NBF dataset and DeepFashion dataset. We compare our results with other state-of-the-art methods.

Table 4.6: Comparison of the proposed method BAFS with other comparative methods

Method	mAP
Baseline Methods	
1. VGG16 - FC7	19.30
2. Alexnet - FC7	25.22
R-MAC [35]	30.03
Rapid-Clothing [151]	33.17
Visual-Search@Pinterest [146]	37.11
Proposed BAFS method	53.60

Figure 4.11: Qualitative comparison of retrieval results of various methods. Retrieved relevant and irrelevant images are shown with green border and red border respectively.

A. Results on NTUBrandFashion (NBF) Dataset

Table 4.7 compares the performance of different methods in terms of mAP (instance search) and NDCG@20 (recommendation). In terms of mAP, the naive triplet loss achieves 33.8%, while the multitask network (triplet+attribute) achieves 56.4%.

The proposed method supervises the triplet loss using the predicted attributes and hence ASML-SW and ASML-SM achieve the best performance with mAPs of 63.79% and 63.71% respectively. Our method clearly outperforms other baselines and competing methods. The works in [151] and [146] used fully-connected layer features after fine-tuning for classification and achieve mAP of 33.1% and 37.7% respectively. The R-MAC [35] descriptor uses features from maximum activation of the final convolutional layer and obtains an mAP of 30%. These methods do not use metric learning, instead they use descriptors from deep CNN networks to rank the images. The VisNet [26] and WTBI [10] use triplet-based and Siamese-based

Table 4.7: Performance comparison of the proposed ASML framework for NTUBrandFashion dataset

Method	mAP (%)	NDCG@20
Baseline Methods		
1. VGG16 - FC7	19.30	56.15
2. Alexnet - FC7	25.22	57.58
Triplet Loss [37]	33.38	69.92
Multitask (Triplet+Attribute)	56.41	76.71
R-MAC [35]	30.03	70.70
Rapid-Clothing [151]	33.17	70.78
Visual-Search@Pinterest [146]	37.11	72.11
VisNet [26]	34.01	63.76
WTBI [10]	41.73	56.43
FashionNet [11]	50.14	80.63
BAFS	53.60	72.27
Proposed ASML-SW	63.79	85.12
Proposed ASML-SM	63.71	83.66

metric learning respectively, where the primary aim is to perform the instance search. They achieve mAPs of 34% and 41.7% respectively. FashionNet [11] jointly uses three types of losses, namely, classification, attribute and triplet loss to train the network and obtains an mAP of 50.1%. This method, however, does not consider the possible interactions between the multiple tasks. In contrast, the proposed attribute-supervised triplet network uses multitask network that not only samples informative triplets but also trains the network based on the triplet importance. Hence, it achieves superior performance to other state-of-the-art methods.

A similar trend can be observed for NDCG in Table 4.7. The proposed attribute-supervised SW and SM triplet network achieve NDCG@20 of 83.66% and 85.12% respectively. Our method clearly outperforms other state-of-the-art methods which demonstrates the advantage of learning a similarity metric space at different granularities.

B. Results on DeepFashion Dataset: Table 4.8 shows the performance of the proposed attribute-supervised triplet network and compares it with the state-of-the-art methods. For all the methods, we list top- k retrieval accuracy, $k = \{1, 10, 20, 30, 40, 50\}$. The proposed ASML-SW and ASML-SM methods achieve top-20 retrieval accuracies of 93.4% and 93.3% respectively. Similar to NTUBrand-Fashion dataset, we can observe that the proposed method outperforms the naive

Table 4.8: Performance comparison of the proposed ASML framework for DeepFashion dataset.

Method / k	1	10	20	30	40	50
VGG16 - FC7	20.2	41.6	48.6	52.9	56.1	58.5
Alexnet - FC7	24.1	45.2	52.2	56.12	58.9	61.2
Triplet Loss	44.9	74.5	81.4	85.0	87.0	88.6
Multitask (Triplet+Attribute)	62.5	87.5	91.5	93.3	94.5	95.3
WTBI [10]	35.6	47.0	50.6	52.0	53.0	54.4
DARN [129]	38.3	61.0	67.5	70.0	71.1	71.8
FashionNet [11]	53.3	73.0	76.4	77.0	79.0	80.0
Batch-All Triplet [123]	59.1	84.1	88.7	91.1	92.2	93.1
Proposed ASML-SW	67.4	89.5	93.4	93.4	95.7	96.3
Proposed ASML-SM	67.9	89.7	93.3	94.8	95.7	96.4

triplet loss and multitask (triplet+attribute) method.

We also compare our proposed method with various baselines and competing methods: WTBI [10], DARN [129], and FashionNet [11]. The WTBI method [10], DARN [129] and FashionNet [11] achieve top-20 accuracies 50.4%, 61.2%, 76.4% respectively. Compared to these, our proposed method achieves superior performance over the state-of-the-art. This shows the advantage of supervising the triplet loss with attribute information during the multitask learning. By treating the triplets based on their importance, the network learns the subtle differences between hard-negative pairs of images, which helps to retrieve the relevant images at higher ranks.

Figure 4.12 shows sample retrieval results with query images and top-5 retrieved image using WTBI, FashionNet, and the proposed method. Exact same instances of clothing are highlighted with green borders. The proposed method retrieves the same items as the query at higher ranks when compared to other methods. From the viewpoint of recommendation, the proposed method retrieves more semantically similar images to the query. For example, in Figure 4.12(b), the proposed model first retrieves images of the same item, followed by items which have the similar model, then brand with slight variation in design and color.

Figure 4.13 and Figure 4.14 show additional retrieval examples from DeepFashion dataset and NBF dataset respectively. From the figures, it can be observed that the proposed method is able to handle various scenarios. For example, the query image in the first row of Figure 4.13 is an image of a dark grey-colored tank top with certain graphic prints. The proposed method first retrieves the same instance

Figure 4.12: Sample qualitative results with query images and their respective top-5 retrieved images with WTBI [10], FashionNet [11] and our proposed method. Exact same instance matches are highlighted with green borders. Best viewed in color.

of the query image from the database, then it ranks other visually similar images which are also clothing images of grey tank tops according to their similarity. We can observe a similar trend in retrieval results for other queries as well.

4.6 Summary

This chapter studied deep learning methods for visual fashion search. In order to capture essential elements of fashion, we first constructed a new NTUBrandFashion (NBF) dataset with 10K fashion images richly annotated with brand logo, attribute, and category information. We proposed a new brand-aware fashion search (BAFS) framework that explores the brand information during the visual search. We demonstrated that the brand information is important in fashion and incorporating it into search framework can significantly improve the fashion search performance.

We also studied deep metric learning method for visual fashion representation and search. We presented a new attribute supervised metric learning framework (ASML) which explores different levels of triplet importance during training. The method uses multitask CNN which shares the mutual information among the task

Figure 4.13: Retrieval examples of the proposed method from the DeepFashion dataset. Images in the first column are query images with their top-5 retrieved images in the respective rows. Correctly retrieved images from the same product ID are highlighted with green borders.

for better tuning the loss. Depending on the significance of the samples, the proposed method first mines informative triplets, and then uses them to train the triplet loss in a soft-manner. The designed framework is able to learn similarity at different granularities, which is essential for visual search as well as recommendation. The proposed method achieves superior performance than other comparative methods on two fashion search datasets. Our method is simple yet effective, thus can be readily adapted for various domain applications.

Figure 4.14: Retrieval examples of the proposed from the NTUBrandFashion dataset. Images in the first column are query images with their top-5 retrieved images in the respective rows. Correctly retrieved images of the same instances are highlighted with green borders.

Chapter 5

Visual Fashion Forecasting

5.1 Introduction

In the previous chapter, we proposed image representation methods for visual fashion search which are independent of time (sales/transaction time). The learned image representation are used for retrieving similar clothing images to perform fashion search. In this chapter, we will develop the time-variant feature representation for fashion images that takes the sale events into account and apply it for visual fashion forecasting.

The main objective of this work is to predict the future popularity of the fashion clothing based on the visual information in fashion images. Fashion images contain rich information that directly relates to attractiveness and popularity of fashion items [162]. We will develop models to learn such visual representation and use it for the fashion trend analysis. The task involves time-variant visual feature representation capturing both visual and trend (temporal) information. In short, this work links the visual analysis and time-series modeling, and predict the future fashion popularity.

Fashion plays a critical role in rapidly growing e-commerce, online shopping, and social media [163, 164]. Analyzing visual fashion trend and predicting the future demand can be immensely valuable for various aspects of fashion industries including designing new clothing, and managing production and inventory. Currently, such analysis requires professional fashion experts [165]. This has two major drawbacks. First, the task is very human intensive and laborious. Second, meaningful insights may not be derived as there can be millions of fashion transactions in

a single day. Hence, we aim to develop a data-driven approach to perform such analysis. We will propose deep learning methods for visual fashion analysis and trend modeling.

In general, the task of fashion forecasting is challenging and requires time-series analysis. A time-series is a sequence of observations x_t which are recorded at each time step t . In fashion domain, the fashion trend changes frequently and products may not be available/sold all the time. For example, consider a brand selling product X in a store. Depending on its sales number or stock availability, it may sell for different periods of time. If the product is not popular or out-of-stock, it will be replaced by another product Y quickly. Therefore, the data (sale number/popularity metric) for a particular fashion item may be incomplete (or even sparse) for a period of interest. Hence, direct time-series modeling cannot be applied to our problem of fashion forecasting. Moreover, fashion images demonstrate a huge diversity in visual appearance. Fashion forecasting should be able to capture the diverse fashion concepts while modeling the trend information. All these pose significant challenges to fashion forecasting.

In recent literature, visual fashion analysis has received growing attention in computer vision community. Fashion clothing retrieval has been studied in [10, 11, 32, 149]. The street2shop methods [11, 116, 150] are proposed to match the clothing images captured from the streets to an online store. Clothing attributes have been used to recognize and describe fashion images [148, 153, 154]. Clothing attributes (such as *red-color*, *checkered-pattern*, etc.) have been employed in fashion analysis. Clothing images have also been used to classify people's styles (e.g. *Goth*, *Hipster*) [145] and for predicting people's occupations [166].

Despite various studies in fashion analysis, only a few works have been carried out to analyze the fashion trend. Fashion forecasting has been studied in [167–169] which used different statistical techniques and neural networks. However, these methods do not incorporate visual information, which is a crucial aspect of fashion. On the other hand, the work in [170] studied the frequencies of clothing attributes in street fashion images against time to analyze their trends. By observing the fashion images from New York Fashion Week (NYFW) event, the authors in [171] studied trends for style elements such as *color*, *pattern*, etc. In the same spirit, [172] studied the influence of New York Fashion show on street fashion in New York. The work in [173] used transaction history from *Taobao* shopping site to mine the best selling clothing attributes for winter and spring seasons. However, it is worth noting that all of these methods analyze the fashion trends in the past.

As opposed to them, our goal is to predict the trend in the future. Recently, [41] proposed a method for forecasting the future popularity of fashion styles. However, the method predicts the future popularity for a limited number of discovered styles, and does not handle individual item forecasting. Moreover, the method uses traditional forecasting method, and is not end-to-end trainable.

In view of this, we aim to develop an end-to-end trainable model based on deep learning techniques to forecast the popularity of individual clothing item. To achieve this, we develop an image-based model and a sequence-based model for fashion forecasting. The image-based model uses raw fashion images, and analyze their features to perform forecasting. The sequence-based model jointly learns the visual representation and trend information to perform forecasting. We will also propose a method based on [41] to perform individual item forecasting. To the best of our knowledge, there is no prior published work on individual clothing item forecasting using visual analytics. Our proposed methods are first-of-its-kind in this topic.

The rest of the chapter is organized as follows. Section 5.2 introduces the problem and describes *clickrate*, a metric to measure fashion popularity. Section 5.3 presents attribute prediction for fashion images which is used in the proposed forecasting models. Section 5.4 describes three proposed forecasting models namely, Style2CR, Image2CR and Seq2CR models. Experiments and results are discussed in Section 5.5. This is followed by a summary in Section 5.6.

5.2 Problem Formulation

Given fashion clothing images with corresponding popularity metric over a time period, our task is to predict the future popularity of individual fashion items at the next time step. The forecast items may be old *i.e.* observed in the past or newly launched items. As highlighted in Section 5.1, the popularity metrics of individual items may be sparse over the period of interest. In other words, there are only few observations for popularity metric for a particular clothing item over the period of interest.

First, we will explain the fashion popularity metric used in this work—*clickrate* (*CR*) which will be used throughout the chapter. We will use an online shopping site to explain the use of clickrate as a fashion popularity metric. Often an online shopping site displays a number of clothing images along with related information

Figure 5.1: Example screen-shot from *Amazon* online shopping site.

such as price. Figure 5.1 shows a sample screen-shot from *Amazon* shopping site for dresses. During a browsing session, a user views and clicks on a particular item when he/she wants to find out more details or purchase the item. Intuitively, the greater the number of the clicks on an item, the more popular the item is. To take into account the number of times an item is shown to users, the number of clicks is normalized by exposure (number of times an item is displayed to users) to obtain *clickrate* of the item *i.e.*, $CR = \text{clicks}/\text{exposure}$. In this work, we will use clickrate as the metric for measuring fashion item popularity. Such click-based metrics have also been used in social media websites such as *Pinterest* [146] and other online shopping companies [26].

5.3 Attribute Prediction

Clothing attributes are instrumental in describing fashion images [11, 32, 145, 153, 170]. In this work, we will use the attribute information for developing the fashion forecasting models. We train an attribute prediction network for fashion images. We obtained dataset of clothing images with annotated attributes from a fashion company. Such attribute labels provide a rich description of fashion items.

We use 194 clothing attributes to train the attribute prediction network. Let

$\mathbf{a} = [a_1, a_2, \dots, a_i, \dots, a_{N_{attr}}]$ be the image attributes, where $a_i \in \{0, 1\}$ denotes the respective binary label. In order to predict the attributes, we design a CNN network where the convolutional layers and fully connected layers are based on VGG16 [24]. We replace the final layer of VGG16 network to match the number of output attributes, and add a sigmoid layer to squash the outputs to be between 0 and 1. The attribute prediction network is then trained using the following binary cross-entropy loss,

$$L_{attr} = - \sum_{i=1}^{N_{attr}} [a_i \log(p_i) + (1 - a_i) \log(1 - p_i)], \quad (5.1)$$

where N_{attr} is the total number of attributes, a_i is the target attribute label, and p_i is the predicted value. The trained network is then used for generating attribute labels for fashion images, which will be used in subsequent forecasting models.

5.4 Visual Fashion Forecasting Models

In this section, we will propose 3 models to predict the fashion popularity in terms of clickrate prediction. Figure. 5.2 shows an overview of different models proposed in this chapter. First, we describe the Style2CR model which is built on top of the method in [41] to predict the clickrate of individual items. This method operates on the attribute information obtained from the attribute prediction network. The attributes are transformed into styles, which are then used to forecast the clickrate. Second, we present an end-to-end trainable Image2CR model which uses raw fashion images to learn the clickrates based on the visual appearance of clothing images. Finally, we describe a sequence-based Seq2CR model, which uses convolutional and recurrent networks to capture the visual and trend information. The following sections describe these three models in detail.

5.4.1 Style-to-CR (Style2CR) Model

In this section, we propose the Style2CR model, which is derived from the work in [41]. This work represents a fashion outfit as a mixture of styles and a style as a distribution of attributes. The key idea is to use the attribute information from fashion images, and transform them into K styles (K is typically 30). Next, it forecasts the popularity of each of these styles. As the work in [41] forecasts the style popularity, it cannot be used to perform individual item popularity forecasting. Therefore, in this section, we will propose modifications to the method

(a) Style2CR Model

(b) Image2CR Model

(c) Seq2CR Model

Figure 5.2: Overview of different forecasting models.

to predict the popularity (clickrate) of individual items. In the following, we will first describe the method in [41], and then present the modification to adapt it for individual clickrate forecasting.

Let $\mathbf{a}_i$ be the predicted attribute vectors for image I_i using the attribute prediction network. All the attributes of the images are collected into an attribute matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ matrix where M is the total number of attributes and N is the total number of images in the dataset. In order to transform the attribute into K styles, the matrix $\mathbf{A}$ is factorized using non-negative matrix factorization (NMF), *i.e.* $\mathbf{A} \approx \mathbf{WH}$. $\mathbf{W} \in \mathbb{R}^{M \times K}$ is the weight matrix which describes the contribution of each attribute for K style. $\mathbf{H} \in \mathbb{R}^{K \times M}$ is K latent style representations for M images. In other words, it can be described by a combination of weights in $\mathbf{h}_k$ (columns in $\mathbf{H}$) *i.e.* each component $h_{k,i} = p(s_k|I_i)$ represents the strength of style (s_k) for image I_i .

Using the relation $h_{k,i} = p(s_k|I_i)$, all the images and their corresponding clickrates are mapped to the style space. We obtain the mapped clickrate for each style s_k using the following equation,

$$y_k^{(t)} = \frac{1}{\sum_i CR_i^{(t)}} \sum_i p(s_k|I_i) CR_i^{(t)} \quad (5.2)$$

where $\sum_i CR_i^{(t)}$ represents the sum of clickrates of all the item at time t , and $CR_i^{(t)}$ represents the clickrate of an image I_i at time t . After this, we can obtain the time series of clickrate $y_k^{(t)}$ for each k style where $k = 1, \dots, K$.

Next, to predict the clickrate for k -style at $t + 1$ time step, we use exponential smoothing method (ES),

$$y_k^{(t+1)} = \sum_{l=1}^t \left[\alpha(1 - \alpha)^{t-l} y_k^{(l)} \right] + (1 - \alpha)^t y_k^{(0)} \quad (5.3)$$

where α is a smoothing factor, and $y^{(0)}$ is the initial observation.

At this point, we have obtained the forecast value $y_k^{(t+1)}$ for each of the K styles. In order to convert the style prediction to individual item forecasting, we use the weights $h_{k,i}$ which represent the strength of style k in item i . We forecast the clickrate for an item (I_i) at $t + 1$ time step by combining the contribution of K

Figure 5.3: Sample images from *Pinterest* website. Fashion images in the top row have 15 times more *likes* compared to the fashion images in the bottom row [12].

styles weighted by $h_{k,i}$ *i.e.*

$$CR_i^{(t+1)} = \sum_{k=1}^K h_{k,i} y_k^{(t+1)}. \quad (5.4)$$

Besides the exponential smoothing used in (5.3) to forecast the styles, we also use other prediction methods such as *last*, *mean* and *drift*. *Mean* prediction refers to using average value of all past observations. *Last* refers to using the last observed value. *Drift* forecasts the value based on $y^{(t+1)} = y^{(t)} + \frac{1}{t-1}(y^{(t)} - y^{(1)})$. We will use these forecasted clickrates as the baseline in our experiments.

5.4.2 Image-to-CR (Im2CR) Model

In this section, we propose an image-based Image2CR (Im2CR) model which uses visual information from raw fashion images for clickrate prediction. The assumption is that fashion images contain rich visual contents which relate to the attractiveness and popularity of fashion items [162]. The study in [12] shows that visual content in fashion images affect the likeability or popularity of items. For example, Figure 5.3 shows sample fashion images from *Pinterest*. The top 3 fashion images received more *likes* (more than 50) while the bottom images received less *likes* (less than 3). Although people may have different interpretations about the clothing, the statistic shows that the fashion clothing in the top row are liked by many people. This shows that the visual attractiveness of fashion images significantly affect their popularity.

Figure 5.4: Proposed Im2CR model for clickrate prediction

In view of this, we model the visual content in fashion images using convolutional networks to learn the representative features responsible for their popularity. The overview of the Im2CR model is shown in Figure 5.4. It consists of a network with convolutional layers followed by linear layers which output the predicted clickrate. The CNN network in the model is trained to recognize visual concepts which affect the popularity of the fashion items. Compared to the style-based models which uses fixed attributes, the Im2CR model learns to extract the representative features $\mathbf{x}$ required for popularity prediction. The feature $\mathbf{x}$ is then used to predict the clickrate,

$$CR_{\text{predicted}} = \mathbf{W}^{(l)}\mathbf{x} + \mathbf{b}^{(l)} \quad (5.5)$$

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the learned weight matrix and bias vector of the last linear layer. The weights in the network are trained to minimize the squared error in clickrate prediction.

5.4.3 Sequence-to-CR (Seq2CR) Model

In this section, we propose an end-to-end sequence-based model for the clickrate prediction which jointly learns the essential visual representation and trend information. For normal sequence modeling, it is assumed that the time series data of a particular item are available at each time step over the period of interest. However, as explained in Section 5.1, the clickrates for a particular fashion item may be incomplete or sparse. In other words, the clickrates for a particular item may only exist in few time steps. The dataset we obtained from a fashion retail company has a similar property. Figure 5.5 shows a plot between the number of fashion items with respect to the number of months for which the clickrate data

Figure 5.5: Histogram of number of items with respect to the number of the months the clickrates are available. Most the items only sold for once or twice.

are available. We can observe that most of the fashion items only occurred once or twice. Thus, the series for of a particular item is very sparse. Therefore, we will propose a sequence selection method to generate sequences for training the model. The generation method is explained below.

In order to construct the sequences, we use clothing attribute information predicted by the attribute prediction network presented in Section 5.3. To achieve this, the training images (I_i) are passed through the prediction network which produces attribute vectors $\mathbf{a}_i$ where i indicates the item id. We aim to generate the sequences such that each image in a particular generated sequence are closer in attribute space *i.e.* similar to each other. The clickrates for the missing observations of a particular item is approximated by the clickrates of other similar items for which the clickrates are available. To achieve this objective, the images are first clustered using the attribute features. We use K-Means clustering to quantize the attributes into N_c clusters. Next, each fashion item is quantized to the nearest cluster centroid, *i.e.* $I_i \rightarrow \mathbf{a}_i \rightarrow c_{id}$ where c_{id} represents the cluster id. Figure 5.6 shows examples of clusters and representative fashion items in each cluster obtained using the clustering process. From the figure, it is observed that images in each row share similar attribute distribution. For example, the cluster in the first row represents *floral-sleeveless-summer* dresses. It is observed that a sequence consists of images with similar attribute distribution. Based on the clus-

Figure 5.6: Example clusters (each row represents a cluster) and representative fashion images obtained for the proposed sequence generation method in Seq2CR model.

ter ids associated with the fashion items, the images are sampled from a particular cluster at each time step to generate training sequences.

The generated sequences are used to train the model which consists of convolutional and recurrent networks. The proposed architecture for the Seq2CR model is shown in Figure 5.7. The model jointly learns the important visual features of fashion images and temporal trend information. Let $(I^{(0)}, I^{(1)}, I^{(2)}, \dots, I^{(t)})$ be the generated sequence of images using sequence generation method at time $(0, 1, 2, \dots, t)$, and $(CR^{(0)}, CR^{(1)}, CR^{(2)}, \dots, CR^{(t)})$ be the corresponding click-rates. Images $I^{(t)}$ are fed into the CNN networks at every time step t , where all CNNs share the same weight parameters. The CNN encodes the visual information and produces a feature vector $\mathbf{x}^{(t)} \in \mathbb{R}^d$, *i.e.* $\mathbf{x}^{(t)} = \text{CNN}(I^{(t)})$. This feature captures the representative information (e.g. looks, attributes, designs etc.) of fashion items that affect their popularity. After obtaining the sequence of the visual features $(\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(t)})$, they are passed through a recurrent

Figure 5.7: Proposed Sequence2CR model for clickrate prediction.

network.

The recurrent network handles the sequential data using hidden states, which allows information flow from one time step to the next. In our method, we use Long-Short Term Memory (LSTM), a variant of RNN, which is able to handle the vanishing and exploding gradient problem in RNNs [80]. The LSTM models the input feature $\mathbf{x}^{(t)}$ at time step t using cell states $\mathbf{c}^{(t)}$ and hidden state $\mathbf{h}^{(t)}$. These states are regulated by special gates namely, input gate $\mathbf{i}^{(t)}$, output gate $\mathbf{o}^{(t)}$, forget gate $\mathbf{f}^{(t)}$ and gate $\mathbf{g}^{(t)}$. Mathematically, the LSTM cell is updated using the following equations.

$$\mathbf{i}^{(t)} = \sigma(\mathbf{W}_{xi}\mathbf{x}^{(t)} + \mathbf{W}_{hi}\mathbf{h}^{(t-1)} + \mathbf{b}_i) \quad (5.6)$$

$$\mathbf{f}^{(t)} = \sigma(\mathbf{W}_{xf}\mathbf{x}^{(t)} + \mathbf{W}_{hf}\mathbf{h}^{(t-1)} + \mathbf{b}_f) \quad (5.7)$$

$$\mathbf{o}^{(t)} = \sigma(\mathbf{W}_{xo}\mathbf{x}^{(t)} + \mathbf{W}_{ho}\mathbf{h}^{(t-1)} + \mathbf{b}_o) \quad (5.8)$$

$$\mathbf{g}^{(t)} = \tanh(\mathbf{W}_{xc}\mathbf{x}^{(t)} + \mathbf{W}_{hc}\mathbf{h}^{(t-1)} + \mathbf{b}_c) \quad (5.9)$$

$$\mathbf{c}^{(t)} = \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} \odot \mathbf{g}^{(t)} \quad (5.10)$$

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \odot \tanh(\mathbf{c}^{(t)}) \quad (5.11)$$

Here $\mathbf{W}_{xi}$, $\mathbf{W}_{hi}$ and $\mathbf{b}_i$ are the input weight matrix, hidden weight matrix and bias vector of the input gate $\mathbf{i}^{(t)}$. Similar notations are used for forget gate, output gate, cell state. The functions σ and $\tanh$ represent sigmoidal and hyperbolic tangent activations respectively. At each time step t , the input gate $\mathbf{i}^{(t)}$, forget gate $\mathbf{f}^{(t)}$ and output gate $\mathbf{o}^{(t)}$ estimate the amount of information to be retained, erased or read in the current cell state $\mathbf{c}^{(t)}$. The current hidden state $\mathbf{h}^{(t)}$ is obtained from the cell state $\mathbf{c}^{(t)}$ through (5.11).

In order to predict the clickrate at every time step t , we feed the hidden state $\mathbf{h}^{(t)}$ to a linear layer which outputs the predicted clickrate given by,

$$CR^{(t)} = \mathbf{W}_{hy}\mathbf{h}^{(t)} + \mathbf{b}_y \quad (5.12)$$

where $\mathbf{W}_{hy}$ and $\mathbf{b}_y$ are the weight matrix and bias vector of the final linear layer. We train the proposed network in an end-to-end fashion to minimize the loss function in (5.13), which is the mean square error (MSE) in clickrate prediction.

$$L_{\text{MSE}}^{(t)} = \frac{1}{N_b} \sum_{n=1}^{N_b} \left[CR_{n,\text{target}}^{(t)} - CR_{n,\text{predicted}}^{(t)} \right]^2, \quad (5.13)$$

$$L^{(total)} = \sum_t L_{\text{MSE}}^{(t)} \quad (5.14)$$

Here N_b is the number of images in a batch, $CR_{n,\text{target}}^{(t)}$ and $CR_{n,\text{predicted}}^{(t)}$ are the ground-truth and predicted clickrates for item n . The loss $L^{(total)}$ is the sum of $L_{\text{MSE}}^{(t)}$ over all time steps. Back-propagation through time (BPTT) algorithm [174] is used to train the network.

5.5 Experiments

5.5.1 Dataset

In this section, we will describe the dataset used for the study of visual fashion forecasting. We perform experiments on a dataset provided by a fashion retail company. The dataset contains 3 categories of fashion items with images and corresponding clickrates recorded for 13 months. The clickrates are expressed in percentage. For each category, there are around 8K fashion items each month. Table 5.1 summarizes the dataset consisting of three categories, together with the means and standard deviations of clickrates. The clickrate data of a particular item in the dataset is not available over the whole period of interest.

Table 5.1: Overview of the dataset used for fashion forecasting.

Category	#items in first 12 months	#items in 13 th month	Clickrate(%)	
			mean	standard deviation
T-Shirt	90,781	6,660	1.31	1.14
Shirt	87,669	6,630	1.44	1.14
Dress	96,676	7,309	1.63	1.63

5.5.2 Experimental Settings

In this section, we first explain settings for data partitioning for performance evaluation and then describe the settings for network architectures and training of the models.

The task of forecasting such as stock-price prediction [175, 176], demand forecasting [87, 88] involves the prediction of future values given the historical data. Our problem here is to predict the click-rate of fashion items. To evaluate such tasks, the historical data are used for training the models, and the data in the future horizon are used for testing. From our data of 13 months (fashion clothing images and corresponding click-rates), we use the first 12 months of historical data to train the models, and data in 13th months (forecast horizon) to test the model.

We use VGG16 [24] as the CNN network for both the Im2CR and Seq2CR models. We initialized the CNN using a network pre-trained on ImageNet [6]. For the sequence modeling, the CNN network provides features of dimension 4096 to the LSTM cells which has a hidden dimension of 512. To generate the sequence, we cluster the clothing attributes into 100 clusters and sample the images from each time-step based on the cluster ids. For each cluster, 1000 sequences are generated in each batch. We use the mean square error (MSE) as our loss function, and train the network using Adam optimizer with an initial learning rate of 10^{-5} . For Style2CR, we use $K = 30$ styles as recommended in [41]. The algorithms are implemented using PyTorch framework [161].

For performance evaluation, we use Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) [177] as the performance metrics. The MAE and MAPE are defined as follows.

$$MAE = \frac{1}{N_t} \sum_{i=1}^{N_t} |CR_{i, \text{target}} - CR_{i, \text{predicted}}| \quad (5.15)$$

$$MAPE = \frac{1}{N_t} \sum_{i=1}^{N_t} \frac{|CR_{i, \text{target}} - CR_{i, \text{predicted}}|}{CR_{i, \text{target}}} \times 100\% \quad (5.16)$$

Figure 5.8: Attribute prediction results on example fashion images. Correctly predicted attributes are highlighted in green.

Here N_t is the total number of test images, $CR_{i, \text{target}}$ and $CR_{i, \text{predicted}}$ are the ground-truth clickrate and predicted clickrate for item i .

5.5.3 Results and Discussion

In this section, we first report the performance of the attribute prediction network presented in Section 5.3. Next, we will present the results and analysis for clickrate prediction using different forecasting models presented in Section 5.4.

A. Attribute Prediction

We used 194 binary attributes to train the attribute prediction network. These attributes incorporate various concepts such as color, patterns, shapes etc. 34K labeled images are used for training while 11K images are used to test the performance of the network. The attribute prediction network achieves a recall of 82.11% with a precision of 83.19%. Figure 5.8 shows examples of attribute prediction in fashion images. From the figure, we can observe that the network is able to handle different types of fashion attributes related to *color*, *pattern*, *shapes* etc. These predicted attributes are used in the Style2CR model to represent styles, and in the Seq2CR model for sequence generation.

B. Clickrate Prediction

The clickrate prediction results for various methods in terms of MAE and MAPE are presented in Table 5.2. We use the Style2CR method as the baseline to compare the performance. The Style2CR-ES is the model that uses exponential smoothing for style forecasting. Similarly, Style2CR-Last, Style2CR-Mean, and Style2CR-Drift denote methods *last*, *mean* and *drift* respectively for style forecasting.

From Table 5.2, it can be observed that all variants of the Style2CR method achieve mediocre performance for all fashion categories. For example, Style2CR-

Table 5.2: Performance of clickrate prediction for various methods on t-shirt, shirt and dress category.

Category Methods	T-shirt		Shirt		Dress	
	MAE	MAPE(%)	MAE	MAPE(%)	MAE	MAPE(%)
Extended Baselines from [41]	1.372	94.8	1.589	95.3	1.789	94.8
	1.376	95.2	1.595	95.8	1.802	95.8
	1.371	94.8	1.588	95.2	1.789	94.8
	1.373	94.9	1.592	95.5	1.794	95.2
Proposed Methods						
Im2CR	0.593	48.9	0.626	44.0	0.716	42.3
Seq2CR	0.541	40.1	0.596	37.5	0.705	42.1

ES obtains an MAE of 1.373 (MAPE of 94.9%) for clickrate prediction in the t-shirt category. The mediocre performance is because the method represent fashion images using style components. The limited number of style representations, however, does not generalize well to individual items. In other words, fashion styles can only model a group of images, but it cannot handle individual item forecasting well.

Our proposed Im2CR model achieves MAEs of 0.593, 0.626 & 0.716 and MAPEs of 48.9%, 44% & 42.3% for t-shirt, shirt and dress category respectively. The model performs significantly better than the Style2CR model. For example, for the t-shirt category, compared to Style2CR-ES, the proposed Im2CR model obtains a relative improvement of 56% in terms of MAE and 47% in terms of MAPE. Unlike Style2CR model, the Im2CR model directly learns representative fashion features to improve the clickrate prediction. This clearly indicates that there exists a strong relationship between the visual appearance of fashion images and their clickrates.

The Seq2CR improves over the Im2CR model by using the fashion trend information. Although the observations for each individual fashion item are not available for the complete period of interest, we mitigate this issue by generating sequences using the proposed sequence generation method. The Seq2CR model achieves the best MAE and MAPE results for all the categories. This demonstrates that the model is able to capture both visual and trend information for clickrate prediction. For instance, for t-shirt category, the Seq2CR model achieves an MAPE of 40.1%, an improvement of 8.8% over the Im2CR model. This clearly shows the advantage of sequence modeling in clickrate forecasting.

5.6 Summary

In this chapter, we studied the problem of visual fashion forecasting which has a huge potential commercial impact. We used clickrate to measure the popularity of the fashion items. In order to predict the clickrate, we proposed an image-based Im2CR model and a sequence-based Seq2CR model. Using the image-based model, we first demonstrated that visual content in fashion images is crucial information in forecasting applications. Next, we developed a sequence generation strategy to handle the non-continuous nature of fashion clickrate data. Using the generated sequences, we trained the Seq2CR model which jointly models the visual content and trend information. The proposed method is able to learn the fashion trend together with visual content of fashion images. We also proposed a method to

extend the work in [41] to perform individual item forecasting. Our experiments show the effectiveness of the proposed methods.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we studied various image feature representation and learning methods for visual search and related applications. Both handcrafted features and deep learning features are explored to learn discriminative features and develop visual search frameworks. The developed frameworks are studied for two selected visual search applications, namely, visual landmark search, and visual fashion search. We used local handcrafted feature based representation for landmark search which required low-level features and geometrical analysis of features for repetitive pattern detection in images. For fashion image representation, we proposed methods based on recent deep learning approaches as the available fashion datasets are large enough to effectively train deep networks. Deep learning based image representations have been developed for visual fashion search and fashion forecasting application. The main results of this thesis are summarized as below.

- (i) In Chapter 3, a handcrafted feature-based visual search framework is developed for the landmark search application. We studied the effect of repetitive patterns on the search performance. In order to alleviate the burstiness issue, a new Lattice-Support Repetitive Local Feature (LS-RLF) algorithm is proposed to detect the repetitive patterns in images. The LS-RLF method estimates the underlying lattice structure of patterns and recovers the undetected features. Using this information, an LS-RLF based image representation is proposed to handle the adverse effect of repetitive patterns in the visual search. A new feature repetitiveness similarity (FRS) metric is also proposed. The FRS metric leverages on the information of repetitive

patterns to improve image matching and search performance. We demonstrated that the repetitive patterns are not a nuisance but can be useful in visual recognition. The experiments conducted on three benchmark datasets clearly show that the proposed methods are effective and outperform other comparative methods.

- (ii) Chapter 4 studied the deep learning methods for visual fashion search. A new fashion dataset called NTUBrandFashion (NBF) with 10K fashion clothing images is constructed. The NBF dataset captures the essential fashion elements such as category, attribute, and brand. In order to explore the importance of brand information, a new brand-aware fashion search framework (BAFS) is proposed. We demonstrated that the brand information is important in fashion and incorporating it into search framework can significantly improve the fashion search performance. We also explored deep metric learning methods to learn discriminative and robust features for fashion search. An end-to-end trainable attribute-supervised triplet network (ASML) is proposed to learn the image embeddings. The method uses multitask CNN which shares the mutual information among the tasks to learn similarity at different granularities which is essential for visual search as well as recommendation. The developed method is simple yet effective, and thus can be readily adapted for various domain applications.
- (iii) Chapter 5 studied an interesting problem of visual fashion representation with time-series information for forecasting application. In particular, the correlation between visual information in fashion images and fashion popularity has been studied. Using visual analysis, various models are developed for individual fashion item forecasting in terms of clickrate prediction. An image-based (Im2CR) model and a sequence-based (Seq2CR) model are developed to forecast the clickrate. The Im2CR model uses visual information in fashion images for clickrate prediction. The Seq2CR jointly learns the visual content and fashion trend information. Experiments show the effectiveness of the proposed methods. Such analysis can be helpful to fashion companies for providing insights on new fashion designs as well as managing their product and inventories.

6.2 Future Work

In this section, we present the possible future research directions and recommendations.

1. Incorporation of exogenous factors for fashion forecasting.

In Chapter 5, we introduced the problem of visual fashion forecasting. The main aim of this work is to investigate the impact of visual information on the popularity forecasting of fashion items. However, there exist several exogenous factors that affect the popularity of fashion items. Factors such as seasons, price, discounts, brand etc. can significantly affect the popularity of fashion items. For example, a good quality product offered with a heavy discount often become popular. It will be interesting to study such factors along with visual analysis for fashion popularity prediction. We can use deep architecture such as CNN to perform the visual analysis and use a neural network to learn a latent feature of the exogenous factors. The generated features can be used to train recurrent network such as LSTM [80], GRU [89] to capture the fashion trend. We can expect promising results from such an analysis.

2. New fashion image generation with fashion trend analysis.

Designing new fashion apparels is a challenging task. It requires fashion experts and analysts to design a new clothing that is attractive and trendy. We propose to use the trend analysis of fashion introduced in Chapter 5 to analyze and extract key visual attributes and features that are potentially responsible for the fashion item popularity. This analysis can assist fashion designers in creating new clothing that will be popular. Such information can be further used in Generative Adversarial Networks (GANs) [92] to create new fashion images. The visual attributes obtained using fashion trend analysis can be incorporated into the generator network during the training of GAN network. This can significantly help the fashion designers as they can directly visualize the fashion images that can be popular in future.

Author's Publications

Journal Papers

1. Dipu Manandhar, Kim-Hui Yap, Zhenwei Miao, and Lap-Pui Chau. Lattice-support repetitive local feature detection for visual search. *Pattern Recognition Letters*, 98:123–129, 2017.
2. Dipu Manandhar and Kim-Hui Yap. Feature repetitiveness similarity metrics in visual search. *IEEE Signal Processing Letters*, 24(9):1368–1372, 2017.

Conference Papers

1. Dipu Manandhar, Muhammet Bastan, and Kim-Hui Yap. Tiered deep similarity search for fashion. In *European Conference on Computer Vision*, pages 21–29. Springer, 2018.
2. Dipu Manandhar, Kim Hui Yap, Muhammet Bastan, and Zhao Heng. Brand-aware fashion clothing search using cnn feature encoding and re-ranking. In *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*, pages 1–5. IEEE, 2018.
3. Zhao Heng, Dipu Manandhar, and Kim-Hui Yap. Hybrid supervised deep learning for ethnicity classification using face images. In *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*, pages 1–5. IEEE, 2018.
4. Wen Zhang, Anu Susan Skaria, Dipu Manandhar, Kim-Hui Yap, and Zhenwei Miao. Augmented visual phrase in mobile product recognition. In *Information, Communications and Signal Processing (ICICS), 2015 10th International Conference on*, pages 1–5. IEEE, 2015.

Bibliography

- [1] Dipu Manandhar, Kim-Hui Yap, Zhenwei Miao, and Lap-Pui Chau. Lattice-support repetitive local feature detection for visual search. *Pattern Recognition Letters*, 98:123–129, 2017.
- [2] Dipu Manandhar and Kim-Hui Yap. Feature repetitiveness similarity metrics in visual search. *IEEE Signal Processing Letters*, 24(9):1368–1372, 2017.
- [3] Dipu Manandhar, Kim Hui Yap, Muhammet Bastan, and Zhao Heng. Brand-aware fashion clothing search using cnn feature encoding and re-ranking. In *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*, pages 1–5. IEEE, 2018.
- [4] Dipu Manandhar, Muhammet Bastan, and Kim-Hui Yap. Tiered deep similarity search for fashion. In *European Conference on Computer Vision*, pages 21–29. Springer, 2018.
- [5] Chih Fong Tsai. Bag-of-words representation in image annotation: A review. *ISRN Artificial Intelligence*, 2012.
- [6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, pages 1097–1105, 2012.
- [7] Christopher Olah. Understanding lstm networks, 2015. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- [8] H. Jegou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1169–1176, June 2009.
- [9] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. Visual place recognition with repetitive structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2346–2359, Nov 2015.

- [10] M Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C Berg, and Tamara L Berg. Where to buy it: Matching street clothing photos in online shops. In *IEEE International Conference on Computer Vision*, pages 3343–3351, 2015.
- [11] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1096–1104, 2016.
- [12] Jinghua Wang, Abrar Abdul Nabi, Gang Wang, Chengde Wan, and Tiansong Ng. Towards predicting the likeability of fashion images. *arXiv preprint arXiv:1511.05296*, 2015.
- [13] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [14] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417. Springer, 2006.
- [15] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *null*, pages 2169–2178. IEEE, 2006.
- [16] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.
- [17] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- [18] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 2241–2248. IEEE, 2010.

- [19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.
- [20] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Ninth IEEE International Conference on Computer Vision*, pages 1470–1477 vol.2, Oct 2003.
- [21] Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 157–166. ACM, 2014.
- [22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.
- [23] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.
- [24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.
- [25] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.
- [26] Devashish Shankar, Sujay Narumanchi, HA Ananya, Pramod Kompalli, and Krishnendu Chaudhury. Deep learning based large scale visual recommendation and search for e-commerce. *arXiv preprint arXiv:1703.02344*, 2017.
- [27] Fan Yang, Ajinkya Kale, Yury Bubnov, Leon Stein, Qiaosong Wang, Hadi Kiapour, and Robinson Piramuthu. Visual search at ebay. *arXiv:1706.03154*, 2017.
- [28] Liang Zheng, Shengjin Wang, and Qi Tian. Lp-norm idf for scalable image retrieval. *Image Processing, IEEE Transactions on*, 23(8):3604–3617, 2014.
- [29] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 883–890, 2013.

- [30] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *European Conference on Computer Vision (ECCV)*, pages 584–599. Springer, 2014.
- [31] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 806–813, 2014.
- [32] Junshi Huang, Wei Xia, and Shuicheng Yan. Deep search with attribute-aware deep network. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 731–732, New York, NY, USA, 2014. ACM.
- [33] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. From generic to specific deep representations for visual recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 36–45, 2015.
- [34] Joe Yue-Hei Ng, Fan Yang, and Larry S Davis. Exploiting local features from deep networks for image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 53–61, 2015.
- [35] Giorgos Tolias, Ronan Sifre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv:1511.05879*, 2015.
- [36] Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4(3):251–258, 2016.
- [37] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNnet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.
- [38] Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3908–3916, 2015.
- [39] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. Simultaneous feature learning and hash coding with deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

- [40] Jiang Wang, Thomas Leung, Chuck Rosenberg, Jinbin Wang, James Philbin, Bo Chen, Ying Wu, et al. Learning fine-grained image similarity with deep ranking. *arXiv:1404.4661*, 2014.
- [41] Ziad Al-Halah, Rainer Stiefelhagen, and Kristen Grauman. Fashion forward: Forecasting visual style in fashion. *International Conference on Computer Vision*, 2017.
- [42] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.
- [43] Stephen M Smith and J Michael Brady. Susan- a new approach to low level image processing. *International journal of computer vision*, 23(1):45–78, 1997.
- [44] Paul R Beaudet. Rotationally invariant image operators. In *International Joint Conference on Pattern Recognition*, volume 579, page 583, 1978.
- [45] Tony Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.
- [46] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.
- [47] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72, 2005.
- [48] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.
- [49] Vijay Chandrasekhar, Gabor Takacs, David Chen, Sam Tsai, Radek Grzeszczuk, and Bernd Girod. Chog: Compressed histogram of gradients a low bit-rate feature descriptor. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2504–2511. IEEE, 2009.
- [50] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2161–2168. IEEE, 2006.

- [51] Vincent Lepetit, Pascal Laguerre, and Pascal Fua. Randomized trees for real-time keypoint recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 775–781. IEEE, 2005.
- [52] Ian H Witten, Alistair Moffat, and Timothy C Bell. *Managing gigabytes: compressing and indexing documents and images*. Morgan Kaufmann, 1999.
- [53] Yu Gang Jiang, Chong-Wah Ngo, and Jun Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 494–501. ACM, 2007.
- [54] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.
- [55] Herve Jegou, Cordelia Schmid, Hedi Harzallah, and Jakob Verbeek. Accurate image search using the contextual dissimilarity measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):2–11, 2010.
- [56] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.
- [57] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [58] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.
- [59] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Readings in information retrieval*, 24(5):355–363, 1997.
- [60] Ondřej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.
- [61] Ondřej Chum, Andrej Mikulík, Michal Perdoch, and Jiří Matas. Total recall ii: Query expansion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 889–896. IEEE, 2011.

- [62] Giorgos Tolias and Hervé Jégou. Local visual query expansion: Exploiting an image collection to refine local descriptors. Research Report RR-8325, INRIA, July 2013.
- [63] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision (ECCV)*, pages 304–317. Springer, 2008.
- [64] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918, June 2012.
- [65] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [66] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Hand-written digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.
- [67] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [68] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.
- [69] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [70] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [71] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.
- [72] Matthew D. Zeiler and Rob Fergus. *Visualizing and Understanding Convolutional Networks*, pages 818–833. Springer International Publishing, Cham, 2014.

- [73] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. Going deeper with convolutions. *Cvpr*, 2015.
- [74] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
- [75] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.
- [76] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- [77] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.
- [78] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.
- [79] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.
- [80] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [81] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.
- [82] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.
- [83] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

- [84] Martin Wöllmer, Florian Eyben, Stephan Reiter, Björn Schuller, Cate Cox, Ellen Douglas-Cowie, and Roddy Cowie. Abandoning emotion classes-towards continuous emotion recognition with modelling of long-range dependencies. In *Proc. 9th Interspeech 2008 incorp. 12th Australasian Int. Conf. on Speech Science and Technology SST 2008, Brisbane, Australia*, pages 597–600, 2008.
- [85] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Michalis A Nicolaou, Björn Schuller, and Stefanos Zafeiriou. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5200–5204. IEEE, 2016.
- [86] Kai Chen, Yi Zhou, and Fangyan Dai. A lstm-based method for stock returns prediction: A case study of china stock market. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 2823–2824. IEEE, 2015.
- [87] Srayanta Mukherjee, Devashish Shankar, Atin Ghosh, Nilam Tathawadekar, Pramod Kompalli, Sunita Sarawagi, and Krishnendu Chaudhury. Armdn: Associative and recurrent mixture density networks for etail demand forecasting. *arXiv preprint arXiv:1803.03800*, 2018.
- [88] Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. Time-series extreme event forecasting with neural networks at uber. In *International Conference on Machine Learning*, number 34, pages 1–5, 2017.
- [89] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [90] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.
- [91] Hojjat Salehinejad, Julianne Baarbe, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. Recent advances in recurrent neural networks. *CoRR*, abs/1801.01078, 2018.
- [92] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

- [93] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [94] David Berthelot, Thomas Schumm, and Luke Metz. Began: boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.
- [95] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.
- [96] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.
- [97] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018.
- [98] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.
- [99] Scott Reed, Zeynep Akata, Xincheng Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.
- [100] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2018.
- [101] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. Towards the automatic anime characters creation with generative adversarial networks. *arXiv preprint arXiv:1708.05509*, 2017.
- [102] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

- [103] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017.
- [104] Chris Donahue, Bo Li, and Rohit Prabhavalkar. Exploring speech enhancement with generative adversarial networks for robust speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5024–5028. IEEE, 2018.
- [105] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.
- [106] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *European conference on computer vision*, pages 392–407. Springer, 2014.
- [107] Eva Mohedano, Kevin McGuinness, Noel E O’Connor, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Bags of local convolutional features for scalable instance search. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 327–331. ACM, 2016.
- [108] Mattis Paulin, Matthijs Douze, Zaid Harchaoui, Julien Mairal, Florent Peronin, and Cordelia Schmid. Local convolutional features with unsupervised training for image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 91–99, 2015.
- [109] Amaia Salvador, Xavier Giró-i Nieto, Ferran Marqués, and Shin’ichi Satoh. Faster r-cnn features for instance search. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on*, pages 394–401. IEEE, 2016.
- [110] Wenhui Jiang, Zhicheng Zhao, Fei Su, and Anni Cai. Aladdin: A locality aligned deep model for instance search. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 2727–2731. IEEE, 2016.
- [111] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744, 1994.

- [112] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 539–546, 2005.
- [113] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Computer Society Conference on Computer vision and pattern recognition*, volume 2, pages 1735–1742, 2006.
- [114] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.
- [115] Sean Bell and Kavita Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 34(4):98, 2015.
- [116] Xi Wang, Zhenfeng Sun, Wenqiang Zhang, Yu Zhou, and Yu-Gang Jiang. Matching user photos to online products with robust deep features. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 7–14. ACM, 2016.
- [117] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Discriminative deep metric learning for face verification in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1875–1882, 2014.
- [118] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(Mar):1109–1135, 2010.
- [119] Faqiang Wang, Wangmeng Zuo, Liang Lin, David Zhang, and Lei Zhang. Joint learning of single-image and cross-image representations for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1288–1296, 2016.
- [120] Qi Qian, Rong Jin, Shenghuo Zhu, and Yuanqing Lin. Fine-grained visual categorization via multi-stage metric learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3716–3724, 2015.
- [121] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4004–4012. IEEE, 2016.

- [122] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *British Machine Vision Conference*, volume 1, page 3, 2016.
- [123] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv:1703.07737*, 2017.
- [124] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv:1411.7923*, 2014.
- [125] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014.
- [126] Xiaofan Zhang, Feng Zhou, Yuanqing Lin, and Shaoting Zhang. Embedding label structures for fine-grained feature representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1114–1123, 2016.
- [127] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, and Yi Yang. Improving person re-identification by attribute and identity learning. *arXiv:1703.07220*, 2017.
- [128] Sameh Khamis, Cheng-Hao Kuo, Vivek K Singh, Vinay D Shet, and Larry S Davis. Joint learning for attribute-consistent person re-identification. In *European Conference on Computer Vision*, pages 134–146. Springer, 2014.
- [129] Junshi Huang, Rogerio S Feris, Qiang Chen, and Shuicheng Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *International Conference on Computer Vision*, pages 1062–1070, 2015.
- [130] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [131] Johannes L Schonberger, Filip Radenovic, Ondrej Chum, and Jan-Michael Frahm. From single image query to detailed 3d reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5126–5134. IEEE, 2015.
- [132] Thomas Leung and Jitendra Malik. Detecting, localizing and grouping repeated scene elements from an image. In *European Conference on Computer Vision (ECCV)*, pages 546–555. Springer, 1996.

- [133] M. Park, K. Broeklehurst, R. T. Collins, and Y. Liu. Deformed lattice detection in real-world images using mean-shift belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1804–1816, Oct 2009.
- [134] James Hays, Marius Leordeanu, Alexei A Efros, and Yanxi Liu. Discovering texture regularity as a higher-order correspondence problem. In *European Conference on Computer Vision (ECCV)*, pages 522–535. Springer, 2006.
- [135] G. Schindler, P. Krishnamurthy, R. Lublinerman, Yanxi Liu, and F. Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, June 2008.
- [136] P. Döubek, J. Matas, M. Perdoch, and O. Chum. Image matching and retrieval by repetitive patterns. In *20th International Conference on Pattern Recognition*, pages 3195–3198, 2010.
- [137] Ondřej Chum and Jiří Matas. Unsupervised discovery of co-occurrence in sparse high dimensional data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3416–3423. IEEE, 2010.
- [138] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 545–552. MIT Press, 2007.
- [139] Yannis Avrithis and Giorgos Tolias. Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval. *International journal of computer vision*, 107(1):1–19, 2014.
- [140] Liang Zheng, Yi Yang, and Qi Tian. Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1224–1244, 2018.
- [141] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2:331–340, 2009.
- [142] Andrea Vedaldi and Brian Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM ’10, pages 1469–1472, New York, NY, USA, 2010. ACM.

- [143] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.
- [144] Edgar Simo-Serra and Hiroshi Ishikawa. Fashion style in 128 floats: joint ranking and classification using weak data for feature extraction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 298–307, 2016.
- [145] M Hadi Kiapour, Kota Yamaguchi, Alexander C Berg, and Tamara L Berg. Hipster wars: Discovering elements of fashion styles. In *European Conference on Computer Vision*, pages 472–488. Springer, 2014.
- [146] Yushi Jing, David Liu, Dmitry Kislyuk, Andrew Zhai, Jiajing Xu, Jeff Donahue, and Sarah Tavel. Visual search at pinterest. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1889–1898. ACM, 2015.
- [147] Masaru Mizuochi, Asako Kanezaki, and Tatsuya Harada. Clothing retrieval based on local similarity with multiple images. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1165–1168. ACM, 2014.
- [148] Wei Di, Catherine Wah, Anurag Bhardwaj, Robinson Piramuthu, and Neel Sundaresan. Style finder: Fine-grained clothing style detection and retrieval. In *Proceedings of the IEEE Conference on computer vision and pattern recognition workshops*, pages 8–13, 2013.
- [149] Jianlong Fu, Jinqiao Wang, Zechao Li, Min Xu, and Hanqing Lu. Efficient clothing retrieval with semantic-preserving visual phrases. In *Asian Conference on Computer Vision*, pages 420–431. Springer, 2012.
- [150] Si Liu, Zheng Song, Guangcan Liu, Changsheng Xu, Hanqing Lu, and Shuicheng Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3330–3337. IEEE, 2012.
- [151] Kevin Lin, Huei-Fang Yang, Kuan-Hsien Liu, Jen-Hao Hsiao, and Chu-Song Chen. Rapid clothing retrieval via deep learning of binary codes and hierarchical search. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 499–502. ACM, 2015.

- [152] Ju-Chin Chen and Chao-Feng Liu. Visual-based deep learning for clothing from large database. In *Proceedings of the ASE BigData & SocialInformatics 2015*, page 42. ACM, 2015.
- [153] Lukas Bossard, Matthias Dantone, Christian Leistner, Christian Wengert, Till Quack, and Luc Van Gool. Apparel classification with style. In *Asian Conference on Computer Vision*, pages 321–335. Springer, 2012.
- [154] Huizhong Chen, Andrew Gallagher, and Bernd Girod. Describing clothing by semantic attributes. In *European Conference on Computer Vision*, pages 609–623. Springer, 2012.
- [155] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *European Conference on Computer Vision*, pages 685–701. Springer, 2016.
- [156] Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. A baseline for visual instance retrieval with deep convolutional networks. *CoRR*, abs/1412.6574, 2014.
- [157] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. *arXiv:1706.07567*, 2017.
- [158] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014.
- [159] Yu Su and Frédéric Jurie. Improving image classification using semantic attributes. *International journal of computer vision*, 100(1):59–77, 2012.
- [160] Jingjing Chen and Chong-Wah Ngo. Deep-based ingredient recognition for cooking recipe retrieval. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 32–41. ACM, 2016.
- [161] <http://pytorch.org/>.
- [162] Xin Lu, Zhe Lin, Hailin Jin, Jianchao Yang, and James Z Wang. Rapid: Rating pictorial aesthetics using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 457–466. ACM, 2014.
- [163] Financial Times. Online retail sales continue to soar, 2018. <https://www.ft.com/content/a8f5c780-f46d-11e7-a4c9-bbdefa4f210b>.

- [164] C. Baldwin. Online spending continues to increase thanks to fashion sector, 2014.
- [165] Philip Hans Franses and Rianne Legerstee. Do statistical forecasting models for sku-level data benefit from including past expert knowledge? *International Journal of Forecasting*, 29(1):80–87, 2013.
- [166] Zheng Song, Meng Wang, Xian-sheng Hua, and Shuicheng Yan. Predicting occupation via human clothing and contexts. In *IEEE International Conference on Computer Vision*, pages 1084–1091. IEEE, 2011.
- [167] Tsan-Ming Choi, Chi-Leung Hui, Sau-Fun Ng, and Yong Yu. Color trend forecasting of fashionable products with very few historical data. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1003–1010, 2012.
- [168] Wen Gu and Xiaogang Liu. Computer-assisted color database for trend forecasting. In *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on*, pages 1–4. IEEE, 2010.
- [169] Tsan-Ming Choi, Chi-Leung Hui, Na Liu, Sau-Fun Ng, and Yong Yu. Fast fashion sales forecasting with limited data and time. *Decision Support Systems*, 59:84–92, 2014.
- [170] Sirion Vittayakorn, Kota Yamaguchi, Alexander C Berg, and Tamara L Berg. Runway to realway: Visual analysis of fashion. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 951–958. IEEE, 2015.
- [171] Shintami C Hidayati, Kai-Lung Hua, Wen-Huang Cheng, and Shih-Wei Sun. What are the fashion trends in new york? In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 197–200. ACM, 2014.
- [172] KuanTing Chen, Kezhen Chen, Peizhong Cong, Winston H Hsu, and Jiebo Luo. Who are the devils wearing prada in new york city? In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 177–180. ACM, 2015.
- [173] Kuan-Ting Chen and Jiebo Luo. When fashion meets big data: Discriminative mining of best selling clothing features. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 15–22. International World Wide Web Conferences Steering Committee, 2017.

- [174] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
- [175] Hao Li, Yanyan Shen, and Yanmin Zhu. Stock price prediction using attention-based multi-input lstm. In *Asian Conference on Machine Learning*, pages 454–469, 2018.
- [176] A Victor Devadoss and T Antony Alphonnse Ligori. Forecasting of stock prices using multi layer perceptron. *International journal of computing algorithm*, 2:440–449, 2013.
- [177] Arnaud De Myttenaere, Boris Golden, Bénédicte Le Grand, and Fabrice Rossi. Mean absolute percentage error for regression models. *Neurocomputing*, 192:38–48, 2016.
- [178] Zhao Heng, Dipu Manandhar, and Kim-Hui Yap. Hybrid supervised deep learning for ethnicity classification using face images. In *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*, pages 1–5. IEEE, 2018.
- [179] Wen Zhang, Anu Susan Skaria, Dipu Manandhar, Kim-Hui Yap, and Zhenwei Miao. Augmented visual phrase in mobile product recognition. In *Information, Communications and Signal Processing (ICICS), 2015 10th International Conference on*, pages 1–5. IEEE, 2015.