DOCTORAL THESIS

Image Retrieval and Object Detection based on Multiple Categories of Queries

Graduate School of Engineering, Iwate University
Doctoral Course, Design & Media Technology
Haitian Sun

March 2020

Acknowledgement

It is a great opportunity for me to publicly thank those who have been influential during my studies at Iwate University. I am very lucky to be directed by my supervisor, Dr. Takuya Akashi, because of his circumspection, endless support, and deep insights. He always inspires me to try harder, whenever I was disappointed with the ideas or experimental results. I would like to thank Dr. Tadahiro Fujimoto, Dr. Takamitsu Tanaka, Dr. Kouichi Konno, Dr. Naoshi Nakaya, and Dr. Katsutsugu Matsuyama who gave me many useful advises since I started my research. I was fortunate to be a member of the Smart Computer Vision Lab at Iwate University. I gently appreciate Dr. Chao Zhang of the University of Fukui, who inspires me a lot and gives me much help in my research. I kindly acknowledge my fellow students for their help, ideas and support, especiallyMr. Jing Zhang, who conducted part of the experiments of TemplateFree. I appreciatively thank all of my friends. Furthermore, I would thank my girlfriend, Yabin Niu, who accompanies me, facing all the difficulties and sharing all the happiness together. At last, I would thank my parents, who always give me a lot but never ask for any return.

ABSTRACT

Nowadays, images and videos are increasingly popular and appear in people's daily life frequently. Institutes carry a surprisingly large number of images, the number of which is still growing fast. It often occurs that people intend to find their desired elements and factors in the massive images. In this case, the approaches to retrieving the images that contain various visual information and detecting the objects in such images accurately and fast are necessary. Therefore, image retrieval (IR) and object detection (OD) has been studied for decades. Content-based image retrieval (CBIR), also known as content-based visual information retrieval (CBVIR), has achieved noticeable successes in the last decades. The term content represents numerous visual information that can be possibly extracted from images, such as color, texture, and shape, rather than semantic meta-data such as tags or descriptions. The CBIR system requires a query and measures the similarity between the query and each database image to rank the database images. OD aims at locating known instances (objects) in images or image sequences. Common OD detects semantic objects of classes with the preknowledge. One extreme situation of OD is when each semantic training class has only one training image. Such a problem is defined as one-shot OD or template matching, where the image for training is also

named as the template. Another extreme situation is named zero-shot OD, meaning there is no image for direct training. The classifier for zero-shot OD is usually trained by the relationship with other known classes.

The query for IR is an input image, representing visual information and treated as a precise request. The query for IR can be an RGB image, depth image, sketch image, contour image, etc., involving various categories of objects. On the other hand, for OD, the semantic objects of classes are considered as queries in this research.

In order to systematize and find the commonalities of the multiple categories of queries, in this thesis, three of them are discussed, including the two for IR and the one for OD.

1) The image containing one whole-body human for IR. Instead of visual similarity defined by colors, shapes, or textures, this research aims to retrieve images with respect to the visual similarity defined by the human pose. In this framework, all the poses are derived from images, which is inspired by the recent development of 3D human pose reconstruction. Furthermore, to make the retrieval more robust against reconstruction error, a recurrent bidirectional similarity measure named recurrent best-buddies similarity (RBBS) is proposed. Both of the qualitative and quantitative results show the usefulness of this framework, especially the quantitative results evaluated by mean average precision (MAP or mAP) exhibit RBBS is improved by 14.13% compared to the most competitive alternative methods.

2) The simplified drawing with only strokes, named sketch for IR. Sketch-based image retrieval (SBIR) is a popular research field that is to rank database images by comparing the similarity between query sketch and database images. This thesis

proposes to compress binary line drawing (sketch) by approximation automatically, considering that it can be applied to SBIR. Specifically, a sketch contains several strokes, each of which can be segmented into several segments by extracting break-points according to the curvature. The approximation of the segments is recorded for the compression. The experiment reveals that the relationship between a certain pair of segments can be represented by some geometrical functions approximately in a rather low dimension. The proposed compressed representation is not only invariant with respect to rotation, scaling, and translation but can also filter out the noise of wobbly lines in some cases if applying it to SBIR.

3) Product images for OD. Product recognition performs a significant role because of its benefits to the compliant arrangements of stores, which further affects the commercial contracts, customer satisfaction, and sale achievement. Automatic recognition systems have been proposed owing to the high cost of the manual inspection by clerks currently. Because of the difficult collection of product images, the systems are commonly in one-shot cases, in which the training data is template product images actually. However, despite the development of one-shot recognition, the systems rarely utilize special characteristics of products on retail store shelves, and the frequent updating of templates is still challenging. Furthermore, it is considered that the product detection can be the basis of product recognition. In this research, instead of the present workflow, a novel product detection system, named TemplateFree is proposed, which combines product segmentation and zero-shot learning. It detects products on retail store shelves by single store shelf images, i.e., corresponding template product images are not necessary. TemplateFree concentrates on the characteristic that a store shelf can be segmented horizontally into

layers then vertically into products so that each product can be detected according to the segmentation. Double zero-shot deep learning frameworks are employed to improve the segmentation. In experiments, TemplateFree achieves better results than the present method.

Contents

Chapter 1

Introduction

1.1 Purpose

Nowadays, images, audios, and videos are increasingly popular and appear in people's daily life frequently. Institutes carry a surprisingly large amount of images, the number of which is still growing fast. It often occurs that people intend to find their desired elements and factors in the massive images. In this case, the approaches to retrieving the images that contain various visual information and detecting the objects in such images more accurately and fast are necessary. Therefore, image retrieval (IR) and object detection (OD) has been studied for decades.

In IR and OD, there exist various categories of queries. The illustrates of the query can be found in Sect. 1.2.2 and Sect. 1.3.2. The different queries must be tackled by different methods into many usages but have robust relationships with each other. Hence, exploring on queries is important and necessary.

In this thesis, to systemize and research on the queries, three categories of queries for IR and OD are proposed, including two for IR and one for OD, as is stated in Sect. 1.4.

1.2 Image Retrieval

1.2.1 Objective

Content-based IR (CBIR), also known as content-based visual information retrieval (CBVIR), has achieved noticeable successes in the last decades. The term "content" represents numerous visual information that can be possibly extracted from images, such as color, texture, shape, rather than semantic meta-data such as tags or descrip-

Figure 1.1: Some examples of IR. The first and the third rows are CBIR, which are retrieved by the Google image. The second row is sketch-based image retrieval (SBIR).

Figure 1.2: An example of OD. In this case, the products (cigarettes) are detected, illustrated by green, which are regarded as the queries in this thesis.

tions. The CBIR system requires a query and measures the similarity between the query and each database image, in order to rank the database images according to the similarities (see Fig. 1.1). The similarity measure can be based on various image features and descriptors, such as speeded up robust features (SURF) [1], histogram of oriented gradients (HOG) [2], and deep features [3].

1.2.2 Query

The word "query" first appears in computing technology is for information retrievals, such as query language, query string, and web search query. The query for IR is an input image, representing visual information and treated as a precise request. A query for IR can be an RGB image, depth image, sketch image, contour image, etc., involving various categories of objects (see Fig. 1.1).

1.2.3 Development

In the last decades, digital images were commonly retrieved by manually annotated keyword tags. However, such method is cumbersome and can consume a lot of human resource. Also, some images are difficult to be described, but can be described by other images. For these reasons, CBIR becomes an important research field.

Low-level Descriptor

Early researchers capitalized on the color-based IR, such as [4] and [5]. Although applying color features has the advantages, e.g., the low computational cost and high accuracy for retrieving the completely same (or extremely similar) images, color-based retrieval is restricted when different things are in a very similar color

space, or when the same things are in absolutely different colors (e.g., the scenery in the summer and the winter of the same place).

Global shape features, such as HOG [2], edge features [6], and texture features [7] are introduced to CBIR to deal with the disadvantage of color-based methods.

The global features (including all the features mentioned above) cannot tackle the changes of rotation, scaling, and violent transformation, so local featrues are also exploited. e.g., the bag-of-features [8], which learns from bag-of-words in information retrieval. Two typical features of this kinds are SIFT [9] and SURF [1]. It worth pointing out that HOG can be also combined with the bag-of-features structure.

The man-made features are named as low-level descriptors. However, there is still a huge semantic gap between the high-level human perceptions and low-level features.

Similarity Measure

Traditional CIBR applies rigid distance functions, such as Euclidean distance, to some low-level features for similarity measure. The semantic gap also makes influence on the rigid distance functions, resulting in the shortest distance may not be the optimization.

Therefore, there are many attempts on similarity measure by machine learning (e.g. [10, 11, 12]). The features extracted from the images are treated as training data, the machine learning methods predict a probability, which replaces the distance. There are mainly two categories of works: 1) Learning to hashing or compact codes [13]; 2) Distance metric learning [14].

Deep Learning

Deep features can also be utilized in CBIR [3]. On one hand, the deep learning framework can be applied to similarity measure, similar to the non-deep machine learning. On the other hand, it can learn a new representation (or descriptor) for an image, named deep feature.

Application

Many relevant interesting applications have been developed, including art collection [15], medical diagnosis [16, 17], and photograph archives [18], inspired by the capability of CBIR. Furthermore, there exist several CBIR system run by companies on the internet: the earliest application-Query by image content (QBIC) [1], TinEye [2], Google Image [3], Yahoo Image Search [4] and so on.

1.3 Object Detection

1.3.1 Objective

OD aims at locating known instances (objects) in images or image sequences (see Fig. 1.2). Common OD detects semantic objects of classes with the preknowledge. Specifically, people create and annotate a large-scale dataset, which is used to train the classifier. The classifier can classify the image patches, which are sampled from the target image by a certain approach, into the semantic objects of classes. By

[1] http://courses.cs.vt.edu/ cs4624/cache/qbic.htm
[2] https://tineye.com
[3] https://www.google.co.jp/
[4] https://images.search.yahoo.com

means of this framework, face detection [19] and pedestrian detection [20] has been well studied.

One extreme situation of OD is when each semantic training class has only one image. Such a problem is defined as one-shot OD or template matching, where the image for training is also named as template. Instead of pre-training, in template matching, online similarity measurement is usually exploited [21].

Another more extreme situation is named zero-shot OD, meaning there exist not images for training but relevant classes as well as semantic relationships. People rely on the relevant known (seen) classes and the semantic relationships to recognize the unknown (unseen) classes [22, 23].

1.3.2 Query

The semantic objects of classes are considered as queries of OD in this thesis (see Fig. 1.2). In some real applications, the query is specific and unique, such as product detection that detects products only, ignoring other objects. Such applications can have various categories of queries.

An OD system can have generalized categories of queries, which can be the product, car, tree, cat, dog, and so on. The categories can also be specified, e.g., cats can be specified into Abyssinian cat, Burmese cat, Chausie cat, etc.. If specified objects are to be located in some images but there is no the corresponding objects for training, the related objects (belong to the same generalized object) can be used for training (or matching) because of their commonality. E.g., if people may intend to locate Abyssinian cats on an image but do not have the image of Abyssinian cats, they can summarize the commonality of Burmese cat and Chausie cat for the

locating. The problem is zero-shot OD. In zero-shot OD problem, the query can be recognized as the generalized object.

1.3.3 Development

Traditional OD

Traditional OD is similar to traditional CBIR, which measures the similarity between image patches (analogized to the database images) and images belonging to the classes (analogized to many queries, which are also named as queries in this thesis). Therefore, there are three important parts of traditional OD, i.e., feature extraction, similarity measure, and sampling, in which most of the feature extraction and similarity measure can be refered to Sect. 1.2.3. Especially, the best-buddies similarity (BBS) [21] and deformable diversity similarity (DDIS) [24] are proposed for the one-shot OD (template matching).

Sampling is the process of displaying candidate image patches, which can be tackled simply by traveling all over the target image to create the heatmap [2, 21, 24] or by some stochastic algorithms, such as genetic algorithm (GA) [25], particle swarm optimization (PSO) [26], and deterministic crowding (DC) [27].

Deep OD

As the development of convolutional neural networks (CNN), deep learning is also applied to OD and achieves huge improvements. Deep OD can be divided into two kinds, one-stage detector and two-stage detector. Two-stage detectors utilize CNN models as backbones only to conduct feature detection or classification, the

remained parts (sampling [5]) are processed by other methods. One of the representative two-stage detectors is Faster R-CNN [19]. One-stage detectors, such as you-look-only-once (YOLO) [28, 29] and single shot multibox detector (SSD) [30], attach the ROI pooling layers to the tail of the backbone framework, making them in one stage.

1.4 Multiple Categories of Queries

Owing to the various categories and usages of queries, their corresponding methods are also different. In order to systemize and find the commonalities of the queries, in this thesis, three categories of queries are discussed:

1) The image that contains one whole-body human;

2) The simplified drawing with only strokes, named sketch;

3) Product images.

1.4.1 Visual Human Pose Retrieval (VHPR)

Instead of visual similarity defined by colors, shapes, or textures, this research aims to retrieve images with respect to the visual similarity defined by the human pose. In this framework, all the poses are derived from images, which is inspired by the recent development of 3D human pose reconstruction. Furthermore, to make the retrieval more robust against reconstruction error, a recurrent bidirectional similarity measure named *recurrent best-buddies similarity* (RBBS) is proposed. Specifically, the similarity measure between two visual poses is treated as a distance measure

[5] Sampling is also called region of interest (ROI) pooling in deep learning

between two-point vectors, with each point representing one of the reconstructed 3D human pose candidates. Then the similarity measure by the displacement of the query recurs. As a justification, the validity of RBBS is verified in a 1D Gaussian situation. In experiments, an original dataset for the retrieval task is built. Both of the qualitative and quantitative results show the usefulness of this framework, especially the quantitative results evaluated by *mean average precision* (MAP or mAP) exhibit RBBS is improved by 14.13% compared to the most competitive alternative methods.

The recurrent bidirectional VHPR and the corresponding experiments are introduced specifically in Chapter 2.

1.4.2 Sketch Compression for SBIR

SBIR is a popular research field that is to rank database images by comparing the similarity between query sketch and database images. However, the so-called sketches by most SBIR researches turn out to be constructed by strokes, containing binary lines only. I propose to compress binary line drawing (sketch) by approximation automatically, considering that it can be applied to SBIR. Specifically, a sketch contains several strokes, each of which can be segmented into several segments by extracting breakpoints according to the curvature. The approximation of the segments is recorded for the compression. The experiment reveals that the relationship between a certain pair of segments can be represented by some geometrical functions approximately in a rather low dimension. The proposed compressed representation is not only invariant with respect to rotation, scaling and translation but can also filter out the noise of wobbly lines in some cases, if applying which to

SBIR.

The approximate representation based sketch compression for SBIR and the corresponding experiments are introduced specifically in Chapter 3.

1.4.3 TemplateFree: Product Detection

Product recognition performs a significant role because of its benefits to the compliant arrangements of stores, which further affects the commercial contracts, customer satisfaction, and sale achievement. Automatic recognition systems have been proposed owing to the high cost of the manual inspection by clerks currently. Because of the difficult collection of product images, the systems are commonly in one-shot cases, in which the training data is template product images actually. However, despite the development of one-shot recognition, the systems rarely utilize special characteristics of products on retail store shelves, and the frequent updating of templates is still challenging. Furthermore, I consider the product detection can be the basis of product recognition. In this paper, instead of the present workflow, a novel product detection system, named TemplateFree is proposed, which combines product segmentation and zero-shot learning. It detects products on retail store shelves by single store shelf images, i.e., corresponding template product images are not necessary. TemplateFree concentrates on the characteristic that a store shelf can be segmented horizontally into layers then vertically into products so that each product can be detected according to the segmentation. Double zero-shot deep learning frameworks are employed to improve the segmentation. In experiments, TemplateFree achieves better results than the present method.

TemplateFree and the corresponding experiments are introduced specifically in

Chapter 4.

Chapter 2

Recurrent Bidirectional VHPR

[1]This chapter has been published in [31]

2.1 Problem Description & Contributions

In spite of the fact that CBIR is independent with annotation quality and completeness of meta-data, it is still a challenging task for traditional image retrieval methods to deal with geometric information (e.g., to retrieve images by similar human poses), due to the reason that traditional CBIR methods, such as SURF and bag of features (BoF) [32], prefer features to geometric information.

In this chapter, I concentrate on a challenging retrieval task named visual human pose retrieval (VHPR), the goal of which is to retrieve dataset images that are similar in human pose to a query image. Each query and dataset image contains a human. The difference between VHPR and CBIR is shown in Fig. 2.1, in which the goals are compared. With this technique realized, many potential applications in art or sports fields can be involved, in which high-cost depth devices [33] are utilized as the main solution. Such devices are usually limited by measurement environments, cost, and mobile inconvenience [34]. Comparing to the depth devices, VHPR requires only monocular images, which can be easily collected by cameras or search engines. One issue of the problem setting of VHPR is the definition of "similar pose". Actually, there exist high-quality human pose datasets, such as LBP [35] and Human3.6m [36], but none of them define and cluster the "similar pose" geometrically instead of semantic action category. Especially for the evaluation, determining the ground truth can be ambiguous. Certainly, people can manually configure a distance threshold to define "similar", but it will inevitably turn out to be a problem of threshold determination.

Existing works attempt to retrieve images with similar human poses by utiliza-

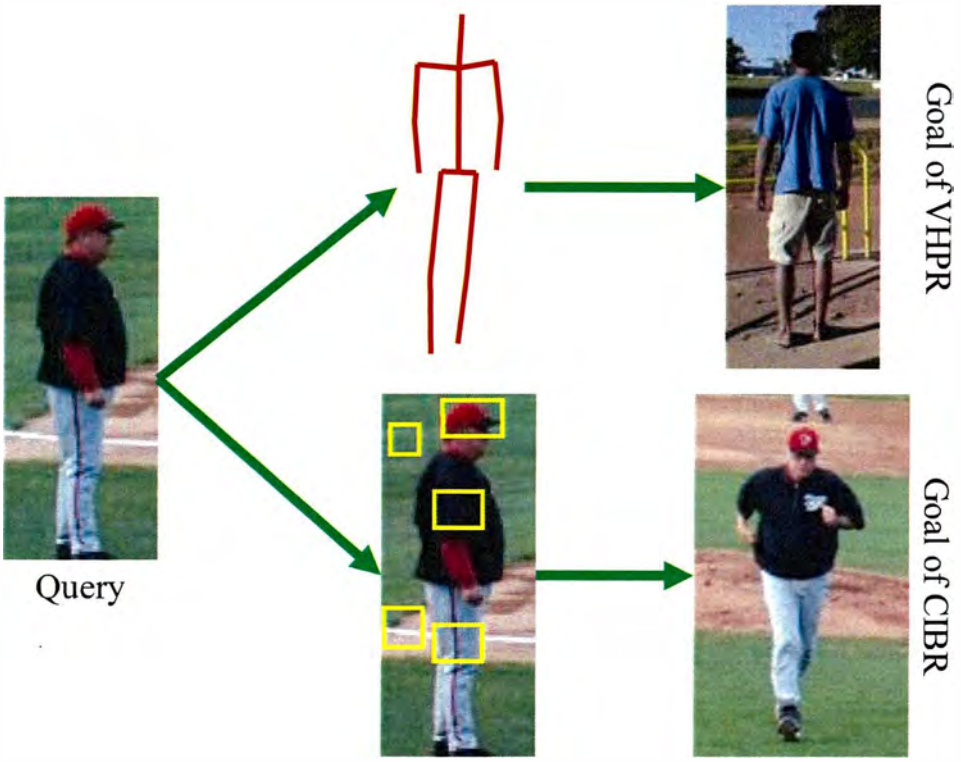


Figure 2.1: Different goals between common CIBR and visual human pose retrieval (VHPR). Common CBIR aims to retrieve images by involving various categories of information, such as the color and texture of dresses and background, while VHPR retrieves images only with the cue of human pose.

tion of 2D human poses [37, 38]. In 2D human pose retrieval, the normalization of scale and angle is difficult, because there exists not camera angle and depth information in 2D cases. Therefore, it works just in case that not only a pair of human poses are similar, but also they are taken at a similar angle and scale.

The reconstructed 3D skeletons is exploited to tackle with the above task. As 3D human poses can be easily normalized to the same direction and scale by rotation and spherical coordinate expression, the limitation of using 2D poses can be overcome. However, human pose reconstruction is an ill-posed problem, which is a problem that has more than one solutions. Despite the existence of ground truth by motion devices, 3D pose reconstruction can have several solutions even marked by human beings owing to the lack of conditions. So far even the recent state-of-the-art research [39] remains a high reconstruction error comparing to ground truth.

In conclusion, issues that need to be settled in VHPR are 1) Ambiguous definition of similar poses; 2) Normalization problem in 2D human pose retrieval; 3) Insufficient accuracy of recovered 3D human poses. Thereby, the main contributions in this section can be concluded as follows:

- A novel dataset composed by 32 classes of distinct poses for evaluation is created.
- This research first proposes VHPR by 3D human pose recovery.
- Instead of the situation that one image can recover one 3D human pose, this research proposes to recover multiple 3D human pose candidates for VHPR.
- RBBS, improved from *best-buddies similarity* (BBS) [21], for the similarity measure of VHPR is proposed.

- RBBS in this research is validated by synthetic experiments.

2.2 Related Work

Performance of VHPR by utilizing 3D human pose as retrieval clue has a close relationship with the confidence of pose extraction. Recovering an articulated 3D human pose from a single image involves two related fields, human pose estimation, and reconstruction, which are two essential cues for solving the VHPR problems in this chapter.

2.2.1 2D Human Pose Estimation

Human pose estimation requires to detect a 2D articulated human pose from a given image. Pedro et al. [40] introduce pictorial structures [41] into 2D human pose retrieval task, which integrates a human body, rather than independent human body parts detection. Although classical hand-crafted image features can be utilized to estimate human pose (e.g. [42]), the recent development of deep learning based frameworks provides more promising solutions. Deep human pose estimation involves developments in framework [43, 44], architecturally refinement [45], and resolution of deformable mixture joints [46]. Such algorithms provide intermediate results from images to 3D human poses.

2.2.2 3D Human Pose Reconstruction

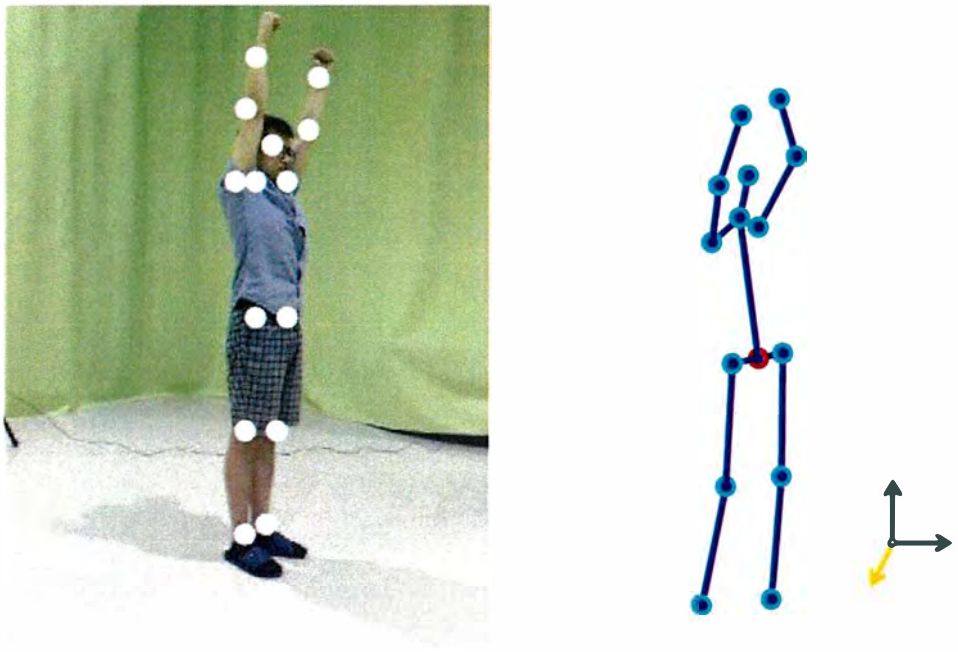
3D human pose reconstruction, also known as 3D human pose estimation or recovery, is to reconstruct a 3D human pose skeleton from a 2D human pose. I only

survey works having a single 2D pose as input rather than multi-view poses. Methods of 3D human pose reconstruction have gone through for periods of assuming the limb length [47], additional input [48, 49, 50, 51, 52], and sparse representation [53, 54, 55]. One recent state-of-the-art method proposed by Chen et al. [56] suggests to predict 3D human poses by a simple matching scheme. However, although the reconstruction error decreases, results of [56] are still severe to be applicable in the case of VHPR that requires to measure the similarity between 3D human poses.

2.2.3 VHPR

The earliest research related with VHPR is [37], in which the authors develop a system for retrieving human upper-body poses by descriptors including both locations and orientations of limbs, and compare with a baseline method that retrieves poses by HOG. The limbs are detected by [57], where the authors propose to reduce search space progressively for body parts estimation. Eichner et al. [38] propose a capable method in highly challenging uncontrolled images, which can estimate upper-body poses in a different scale, to improve the human pose estimation, and thereby yield in the improvement of upper-body pose retrieval. These works can only retrieve upper-body poses, while this research attempts to build a retrieval framework with the whole skeleton, which enlarges the degree of freedom and makes the task more challenging. Ren et al. [58] first introduce a retrieval solution with the whole 2D skeleton, while they do not handle with the problem of normalization well and thus their method works only in the case that images are shot at a similar angle and scale.

To the best of my knowledge, there exists no similar research on VHPR that



(a) (b)

Figure 2.2: Skeleton model of N-joint human pose. **(a)** 2D human pose example. **(b)** 3D human pose example. The pose in (b) is reconstructed from (a). The structure of 2D and 3D poses are the same. The joint between hips in (b), illustrated by green, is an assistant joint which does not exist in real for convenience, named "center".

measures similarity by entire 3D human poses so far. In this chapter, I first exploit 3D human poses for VHPR, under the assumption that the results of pose reconstruction with estimation errors.

2.3 From Image to 3D Human Pose Candidates

Final result of the 3D human pose reconstruction algorithm [56] is the first ranked vector of combined 3D points (i.e., each point representing the according position of a joint), denoted by $\mathbf{X}_1 \in \mathbb{R}^{N\times 3}$, where N is the number of articulated joints. In addition to $\mathbf{X}_1$, we keep the lower-ranked matchings as potential candidates to form a set of poses $\mathbf{C} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_i, ..., \mathbf{X}_m | \mathbf{X}_i \in \mathbb{R}^{N\times 3}\}$. In another word, in the retrieval framework, top-m ranked candidates of the 3D poses are utilized as "features" to represent a certain 2D pose. I claim that as 3D reconstruction problem is highly ill-posed, the first ranked output can possibly be a less satisfactory result, due to difficulties such as depth ambiguity [55].

For an input image $\mathbf{I}$, its predicted 2D pose by [44] is denoted in $\mathbf{x}(\mathbf{x} \in \mathbb{R}^{N\times 2})$. Under the assumption given by [56] that the 2D pose estimation is independent with the prediction of $\mathbf{X}_i$ from $\mathbf{x}$, the joint probability w_i can be written as,

$$w_i = \underbrace{\mathrm{p}(\mathbf{X}_i|\mathbf{x})}_{[56]} \cdot \underbrace{\mathrm{p}(\mathbf{x}|\mathbf{I})}_{[44]} \cdot \mathrm{p}(\mathbf{I}). \tag{2.1}$$

By sorting with respect to normalized $w_i \in [0, 1]$, $\mathbf{C}$ can then be determined. I unify the skeleton models of both $\mathbf{X}_i$ and $\mathbf{x}$ ($N = 14$), which means that corresponding joints from $\mathbf{X}_i$ and $\mathbf{x}$ refer to the same joint (i.e., with the same semantic defini-

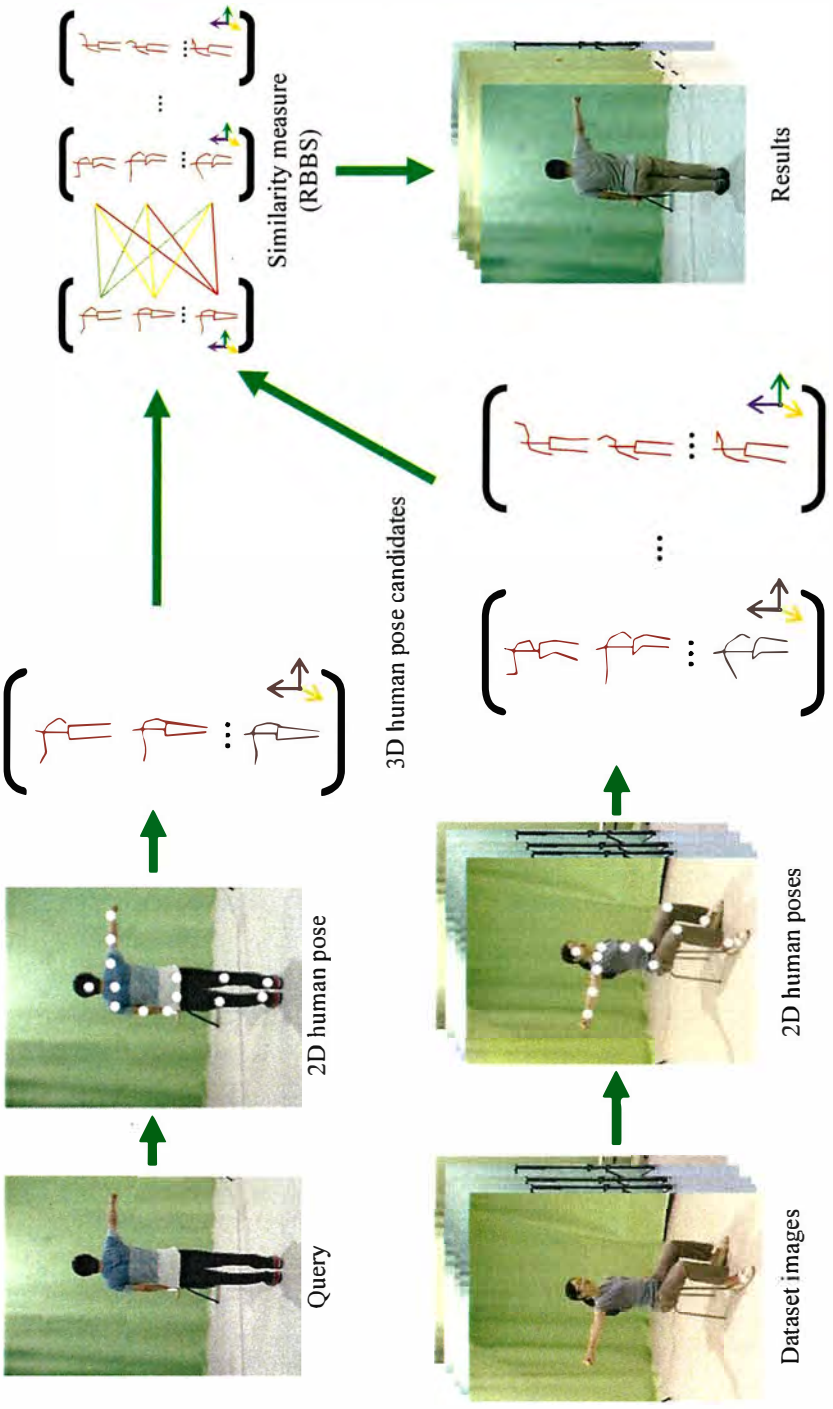


Figure 2.3: The overview of the retrieval framework. 2D poses are directly estimated from the images, with each joint illustrated by a white circle. The 2D poses are reconstructed to m top-ranked candidates for similarity measure. The result images are ranked according to the similarity defined by RBBS.

tion). The skeleton model is shown in Fig. 2.2, including head, neck, and a pair of shoulders, elbows, wrists, hips, knees and ankles. Each w_i is used for calculating weighted BBS in Eq. 2.7, which measures bidirectional similarity and is introduced in Sect. 2.4.2, for it indicates the reconstruction confidence of $\mathbf{X}_i$.

To eliminate the influence brought by the scaling, rotation, and translation during retrieval, we first normalize each $\mathbf{X}_i$ into the same scale according to [54]. Specifically, a mean pose $\bar{\mathbf{X}}$ from CMU motion data [53] is learned first. After then, vectors between each connected two joints $\mathbf{J}_a$ and $\mathbf{J}_b$ can be denoted by spherical coordinates, written as

$$\mathbf{J}_a - \mathbf{J}_b = (\phi_{ab}, \varphi_{ab}, l_{ab}), \tag{2.2}$$

where ϕ_{ab} is the zenith angle, φ_{ab} is the azimuth angle, and l_{ab} is the vector length. Keeping ϕ_{ab} and φ_{ab} invariant, I adjust l_{ab} to the same length of the corresponding connection in $\bar{\mathbf{X}}$. Furthermore, in order to conduct pose normalization (considering rotation & translation), the skeletons are rotated to the same direction. The forward direction $\mathbf{d}$ of a skeleton is computed by

$$\mathbf{d} = \mathbf{v}_{cn} \times \mathbf{v}_s, \tag{2.3}$$

where $\mathbf{v}_{cn}$ is the vector from "center" to the neck, $\mathbf{v}_s$ is the vector from the right shoulder to the left shoulder. The "center" is the assistant joint shown in Fig. 2.2. The "center" also bolster the normalization of translation, where the whole body is moved by the vector from "center" to the origin.

2.4 Approach of Retrieval

2.4.1 Retrieval Framework and Problem Setting

An image retrieval framework includes feature extraction of query and dataset images and similarity measure commonly. In this method, the "feature" can be described as $\mathbf{C}$, as well as the similarity measure algorithm is RBBS, introduced in Sect. 2.4.2. The retrieval framework is shown in Fig. 2.3.

In such a retrieval framework, 3D human pose recovery and the calculation of RBBS are key steps. I consider each $\mathbf{X}_i \in \mathbf{C}$ as a dimension, which is usually a number of feature vectors. Therefore, to measure the similarity between two visual poses can be converted to matching two point vectors, the dimension of which is m, and the dimension of each high-dimensional point $\mathbf{X}_i$ is $N \times 3$. I denote a series of similarities measured by RBBS between candidates of query $\mathbf{C}^{\mathrm{q}}$ and candidates of dataset images $\mathbf{D} = \{\mathbf{C}_1^{\mathrm{d}}, \mathbf{C}_2^{\mathrm{d}}, ..., \mathbf{C}_n^{\mathrm{d}}\}$ by RBBS($\mathbf{C}^{\mathrm{q}}, \mathbf{D}$), where n is the size of dataset, q and d represent query and dataset image respectively.. Analogously, I denote a series of similarities measured by BBS as BBS($\mathbf{C}^{\mathrm{q}}, \mathbf{D}$). For the similarity between a pair of candidates $\mathbf{X}$ and $\mathbf{Y}$, I denote it by s($\mathbf{X}, \mathbf{Y}$).

2.4.2 RBBS

The original BBS is proposed for template matching [21], which can match a pair of point vectors, and thus BBS can be employed in this research. BBS is defined as the fraction of *Best-Buddies Pairs* (BBPs). The BBP between two vectors $\{p_i \in P\}$

and $\{q_i \in Q\}$ can be defined as

$$\mathrm{bb}(p_i, q_j) = \begin{cases} 1, \ \mathrm{nn}(p_i, Q) = q_j \wedge \mathrm{nn}(q_j, P) = p_i \\ 0, \ \text{otherwise} \end{cases}, \tag{2.4}$$

where $\mathrm{nn}(p_i, Q) = \mathrm{argmin}\{\mathrm{distance}(p_i, q)\}$. In this research, as $\mathbf{X}_i$ represents a high-dimensional point, for $\mathbf{X}_i \in \mathbf{C}_X$ and $\mathbf{Y}_j \in \mathbf{C}_Y$ $(i, j \in [1, m],\ i, j \in \mathbb{Z})$, Eq. 2.4 can be written as

$$\mathrm{bb}(\mathbf{X}_i, \mathbf{Y}_j) = \begin{cases} 1, \ \mathrm{S}(\mathbf{X}_i, \mathbf{C}_Y) = \mathbf{Y}_j \wedge \mathrm{S}(\mathbf{Y}_j, \mathbf{C}_X) = \mathbf{X}_i \\ 0, \ \text{otherwise} \end{cases}, \tag{2.5}$$

where $\mathrm{S}(\mathbf{X}_i, \mathbf{C}_Y) = \mathrm{argmin}\{\mathrm{s}(\mathbf{X}_i, \mathbf{Y}) | \mathbf{Y} \in \mathbf{C}_Y\}$. In other words, BBP is two elements in different vectors, where the nearest neighbor of one element in the opposite vector is another element, and vice versa. BBS between two point sets P and Q is defined in [21] as

$$\mathrm{bbs}(P, Q) = \frac{1}{\min\{M_1, M_2\}} \cdot \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \mathrm{bb}(p_i, q_i), \tag{2.6}$$

where M_1 and M_2 are the length of the two vectors. Since I reconstruct m $\mathbf{X}_i$ for each image, Eq. 2.6 can be written as

$$\mathrm{bbs}(\mathbf{C}_X, \mathbf{C}_Y) = \frac{1}{m} \cdot \sum_{i=1}^{m} \sum_{j=1}^{m} \mathbf{C}_X.w_i \cdot \mathbf{C}_Y.w_j \cdot \mathrm{bb}(\mathbf{X}_i, \mathbf{Y}_j). \tag{2.7}$$

The expression $\mathbf{A}_1.g$ denotes g belongs to $\mathbf{A}_1$.

BBS remains a problem that some of the similarities can be equal, which cannot be ranked. I calculate $\mathrm{s}(\mathbf{X}_1^{\mathrm{q}}, \mathbf{X}_1^{\mathrm{d}})$ for the solution of this problem. In detail, in order

to avoid repetitive computation, in computing $\mathrm{bbs}(\mathbf{C}^{\mathrm{q}}, \mathbf{C}_t^{\mathrm{d}})$, I record $\mathrm{s}(\mathbf{X}_1^{\mathrm{q}}, \mathbf{X}_{1,t}^{\mathrm{d}})$ ($\mathbf{X}_{1,t}^{\mathrm{d}} \in \mathbf{C}_t^{\mathrm{d}}$). Thus the similarity is given by

$$\mathrm{ebbs}(\mathbf{C}^{\mathrm{q}}, \mathbf{C}_t^{\mathrm{d}}) = \mathrm{bbs}(\mathbf{C}^{\mathrm{q}}, \mathbf{C}_t^{\mathrm{d}}) \cdot 10^{\mathrm{om}(MS)+1} + s(\mathbf{X}_1^{\mathrm{q}}, \mathbf{X}_{1,t}^{\mathrm{d}}), \tag{2.8}$$

where $\mathrm{om}(\cdot)$ denote the order of magnitude and

$$MS = \max \mathrm{s}(\mathbf{X}_1^{\mathrm{q}}, \mathbf{X}_{1,t}^{\mathrm{d}}), t \in [1, n], t \in \mathbb{Z}. \tag{2.9}$$

The Eq. 2.8 guarantees BBS is the main similarity, meanwhile, the EBBS can be affected by $\mathrm{s}(\mathbf{X}_1^{\mathrm{q}}, \mathbf{X}_{1,t}^{\mathrm{d}})$. I denote a series of the Euclidean-BBS (EBBS) as $\mathrm{EBBS}(\mathbf{C}^{\mathrm{q}}, \mathbf{D})$.

I further refine the retrieval by recurrence. I deem the second half of $\mathbf{C}^{\mathrm{q}}$ may not similar to the ground truth, especially comparing to the first half of the first rank by EBBS, represented by $\mathbf{C}^{\mathrm{tr}}$. I evaluate the fraction of BBP, denoted by f, which is compared with a manually decided floating number $\gamma \in [0, 1]$. If $\gamma \geq f$, I displace $\mathbf{C}^{\mathrm{q}}$ by $\mathbf{C}^{\mathrm{dq}} = \mathrm{dp}(\mathbf{C}^{\mathrm{q}}, \mathbf{C}^{\mathrm{tr}})$, where

$$\mathrm{dp}(\mathbf{C}^{\mathrm{q}}, \mathbf{C}^{\mathrm{tr}}) = \underbrace{\{\mathbf{X}_1^{\mathrm{q}}, \ldots, \mathbf{X}_{\lceil \frac{m}{2} \rceil}^{\mathrm{q}}, \mathbf{X}_{\lceil \frac{m}{2} \rceil + 1}^{\mathrm{tr}}, \ldots, \mathbf{X}_m^{\mathrm{tr}}\}}_{\mathbf{X}_i^{\mathrm{q}} \in \mathbf{C}^{\mathrm{q}}, \mathbf{X}_i^{\mathrm{tr}} \in \mathbf{C}^{\mathrm{tr}}, i \in [1,m], i \in \mathbb{Z}}, \tag{2.10}$$

is to displace the second half of $\mathbf{C}^{\mathrm{q}}$ by the first half of $\mathbf{C}^{\mathrm{tr}}$. As time complexity of BBS is m^2 times than normal matching (e.g., Euclidean distance), the recurrence is operated by one time.

The algorithm of RBBS is clarified in Algorithm 1 in summary. The retrieval

Algorithm 1 RBBS

Input: $\mathbf{C}^q$, $\mathbf{D}$
  Compute EBBS($\mathbf{C}^q, \mathbf{D}$)
  Sort EBBS($\mathbf{C}^q, \mathbf{D}$) $\Rightarrow$ Acquire top ranking $\mathbf{C}^{tr}$
  Compute $f = \frac{\text{bbs}(\mathbf{C}^q, \mathbf{C}^{tr})}{m}$
  if $f \geq \gamma$ **then**
    Compute $\mathbf{C}^{dq} = \text{dp}(\mathbf{C}^q, \mathbf{C}^{tr})$
    Compute RBBS($\mathbf{C}^q, \mathbf{D}$) = EBBS($\mathbf{C}^{dq}, \mathbf{D}$)
  else
    RBBS($\mathbf{C}^q, \mathbf{D}$) = EBBS($\mathbf{C}^q, \mathbf{D}$)
  end if
Output: RBBS($\mathbf{C}^q, \mathbf{D}$)

result is acquired by the sort of RBBS($\mathbf{C}^q, \mathbf{D}$).

2.4.3 Similarity Between a Pair of Single 3D Poses

Simply, I compute Euclidean distance between a pair of 3D human poses to measure their similarity. A 3D human pose $\mathbf{X} \in \mathbb{R}^{N\times 3}$ can be expressed concretely as $\mathbf{X} = \{\mathbf{J}_1, \mathbf{J}_2, ..., \mathbf{J}_N | \mathbf{J}_k \in \mathbb{R}^3, k \in [1, N]\}$. The similarity between two 3D human poses $\mathbf{X}$ and $\mathbf{Y}$ is measured by

$$\text{s}(\mathbf{X}, \mathbf{Y}) = \sum_{k=1}^{N} ||\mathbf{X}.\mathbf{J}_k - \mathbf{Y}.\mathbf{J}_k||_2. \tag{2.11}$$

2.5 Validity of RBBS in VHPR

The main effect on the precision of VHPR originates from the insufficient accuracy of pose recovery. Focusing on such a characteristic, I analogize poses to generate synthetic data in a low dimension and conduct synthetic experiments, so as to confirm the validity of RBBS in VHPR via more comprehensive experiments.

2.5.1 Low-dimensional Synthetic Data

I build synthetic data in a low dimension because the dimension of the original human pose data (3D) is high. The higher dimension results in the more computation time, so it is severe to conduct experiments in a large dataset. The synthetic data is generated based on Gaussian distribution, rather than random generation. Randomly generated data cannot ensure the data is in similar categories to each other, which can raise the precision of matching by geometry distance owing to the probable obvious distinction.

I summarize VHPR as a problem to match vectors of points in Sect. 2.4.1. A human pose image has its corresponding accurate 3D ground truth, but $\mathbf{X}_i$ from images, employed in VHPR, exist reconstruction error. Such reconstruction error makes the candidates approximated to the ground truth. Accordingly, I present to analogize the original data by approximate and 1D discrete Gaussian distribution.

A 1D Gaussian form can be written as $\mathrm{N}^1(\mu, \sigma)$, where μ is the expectation and σ is the Gaussian radius. The analogy of poses and synthetic data is exampled in Fig. 2.4. Synthetic ground truths are generated via altering σ and keeping μ invariant. I analogize the reconstruction error by randomly adjusting values of the ground truth in a range $[g_i - r, g_i + r]$, where g_i is the value of i th dimension. For the quantity of synthetic data, I generate 100 synthetic ground truths, each of which corresponds to a query and 100 vectors of candidates to be retrieved.

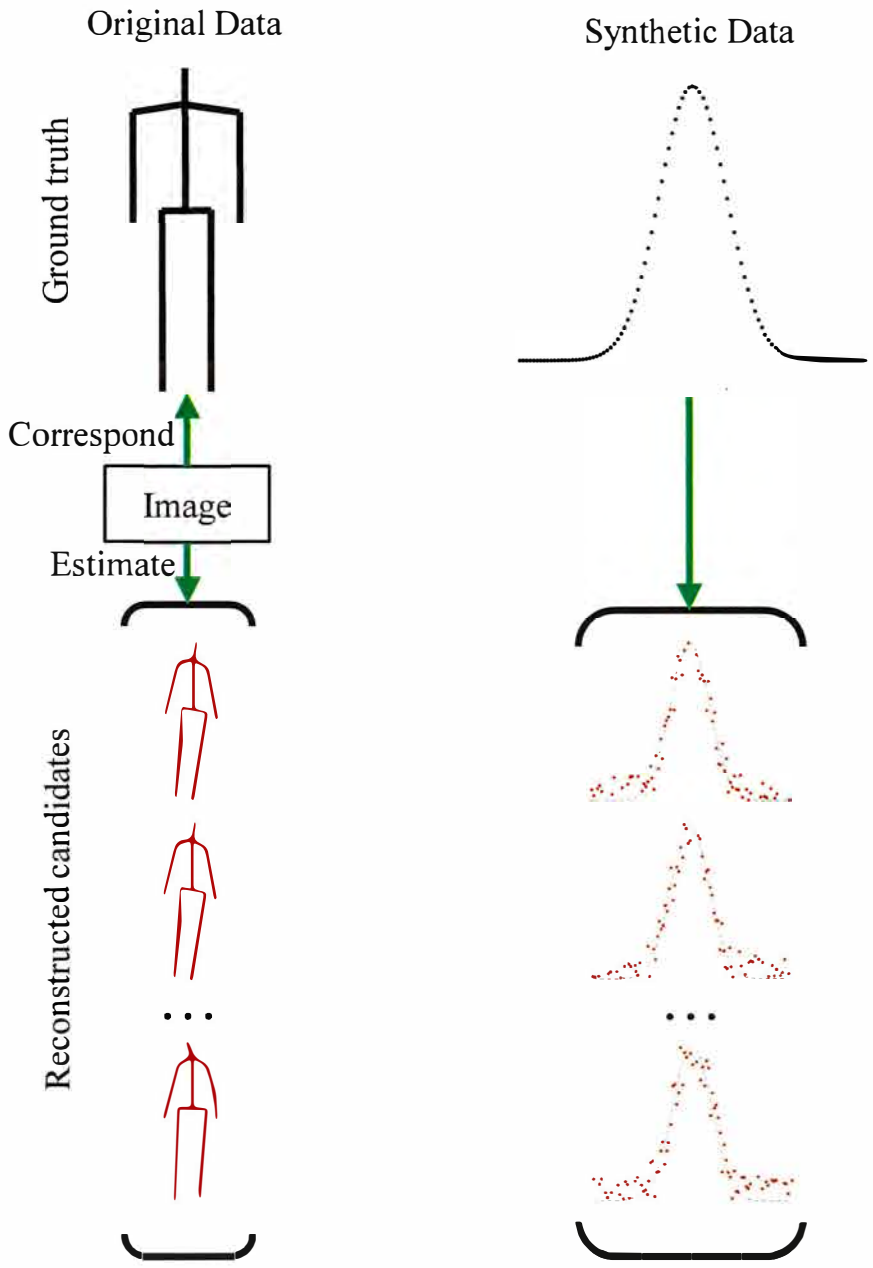


Figure 2.4: Synthetic data generation. In order to gain more insight into the validity of RBBS, I represent the task in 1D Gaussian form to conduct more comprehensive experiments. Synthetic data of ground truth, reconstructed candidates, and original data of reconstructed candidates in the experiments are exampled. The ground truth of the original data is one of the human pose templates in the experiments.

2.5.2 Synthetic Experiments

Evaluation Method

I evaluate the performance by *mean average precision* (MAP or mAP), following [38] and the precision curve. The precision is the fraction of the retrieved relevant images in retrieved images. The precision curve shows the change of mean precision with the number of retrieved image increasing. The precision curve represents a better result if the curve is higher. The MAP for a set of queries is the mean of the average precision scores for each query. Hence precision and MAP are floating numbers within $[0, 1]$, and they indicate better results when they are more closed to 1.

Comparison against Euclidean Distance

I compare RBBS with Euclidean distance. The 1D Euclidean distance here is determined as the sum of absolute value of the distance between a pair of points. Specifically, for a pair of candidates $\mathbf{U}$ and $\mathbf{Z}$, the Euclidean distance is

$$\mathrm{ED}(\mathbf{U}, \mathbf{Z}) = \sum_{i=1}^{N} |u_i - z_i|,\ u_i \in \mathbf{U},\ z_i \in \mathbf{Z}. \tag{2.12}$$

The Euclidean distance of a pair of candidate vectors is calculated by $\mathrm{ED}(\mathbf{U}^s, \mathbf{Z}^s)$, where $\mathbf{U}^s$ and $\mathbf{Z}^s$ are most similar to their corresponding ground truth, imitating the optimal results.

In Fig. 2.5 and Tab. 2.1, I set $r = 0.3$. The results indicate the following conclusions: 1) The performance of RBBS raises with the increase of m; 2) When m is less than a threshold within $(10, 30)$, RBBS performs worse than Euclidean dis-

tance; 3) Overall, RBBS improves the performance in retrieving approximate point sets than Euclidean distance.

2.6 Experiment

2.6.1 Dataset

I create a dataset for both qualitative and quantitative evaluation. In order to involve less ambiguity in the definition of "similar pose", 10 volunteers are asked to imitate 32 predetermined types of poses. Each type of pose is designed to be distinct from the others. As illustrated in Fig. 2.6, dataset images are taken by three evenly placed web cameras on a circle with the volunteer as the center point and 2 m as the radius. The cameras are 1 m from the ground. Diversity in physique and gender of the volunteers is considered.

I design the poses to form the dataset following two rules: 1) distinct from each other; 2) easy to be imitated by the volunteers. First, I verify the poses by either stretching arms forwards, sidewards, upwards, or downwards as stretching limbs in orthogonal directions, which meets both above rules. Further, I add poses of standing and sitting straight up with legs naturally bending. The typical examples of designed poses are shown in Fig. 2.7. The number of designed poses is 32. The dataset totally contains 960 images (32 pose classes $\times$ 10 volunteers $\times$ 3 camera angles). I randomly select 32 queries one by one from each pose class.

Table 2.1: Comparison between RBBS and Euclidean distance by synthetic data.

Method & Parameter		Mean average precision (MAP)
RBBS	m = 10	0.151
	m = 30	0.232
	m = 50	0.268
	m = 100	0.336
	m = 150	0.376
	m = 200	0.399
Euclidean Distance		0.161

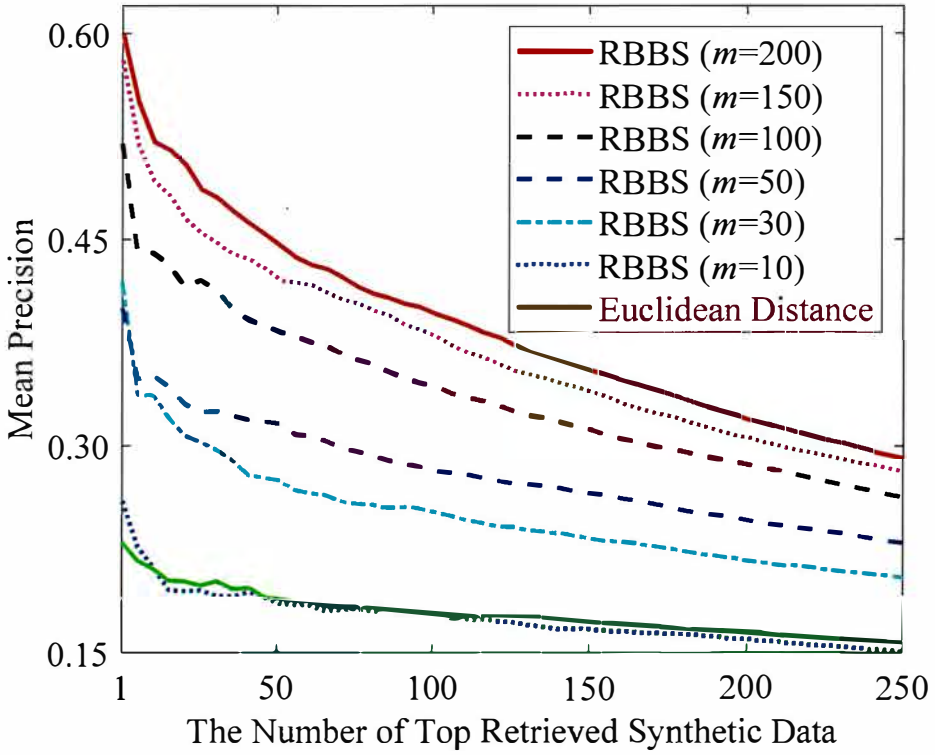


Figure 2.5: Comparison between RBBS and geometry distance by synthetic data (r = 0.3). In synthetic experiments, the performance of RBBS is affected by m and is better than that of Euclidean distance. RBBS performs better with the increase of m.

2.6.2 Effect of the Parameter m

Parameter m appears in the calculation of RBBS, which means the number of candidates recovered from an image. The synthetic experiments suggest that the increase of m can improve the performance of RBBS, while in this section, I explore how the m affects the whole performance in real retrieval task quantitatively. I increase m from 10 to 500 and at the same time evaluate the change of MAP. The result is shown in Fig. 2.8(a), which indicates that 1) The same to Fig. 2.5, the MAP increases overall by the increase of m; 2) Choices around $m = 200$ are suggested as the increase of MAP starts to slow down with respect to the increase of m; 3) RBBS can perform worse than only applying Eq. 2.11 for similarity calculation when m is small, which can similarly be observed in Fig. 2.5. It worth mentioning that involving less confident candidates (e.g., $m > 300$) can involve outliers in similarity measurement, which will reasonably not improve the performance.

Although the increase of m leads to a higher computational cost, I do not conduct the experiments on computational costs, because the computational complexity is computable mathematically. The computational complexity of computing a Euclidean distance between a pair of skeletons is $O(1)$, while the computational complexity of RBBS is $O(m^2)$.

2.6.3 Comparative results

In this section, I compare the proposed method against other alternatives. As the increase of m leads to more computational cost, to make RBBS efficient, I fix $m = 200$ in the following experiments. Specifically, I compare 3D pose+RBBS with

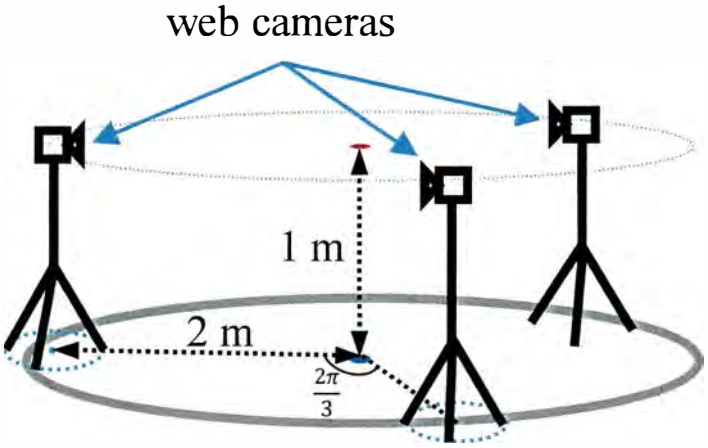


Figure 2.6: Dataset creation system. I represent web cameras by the graphic assembled with a triangle and a rectangle. The web cameras are fixed on the tripods that are uniformly (with an interval of $\frac{2\pi}{3}$) located at a circle with a radius of 2 m. The circle centers at the person who makes poses.

Table 2.2: Comparison against existing methods by MAP.

Method			Mean average precision (MAP)
Euclidean Distance	BoF	SURF	0.037
		HOG	0.037
	HOG		0.054
	2D Pose		0.202
	3D Pose		0.269
RBBS	3D Pose	$m = 200$	0.269
		$m = 500$	0.307

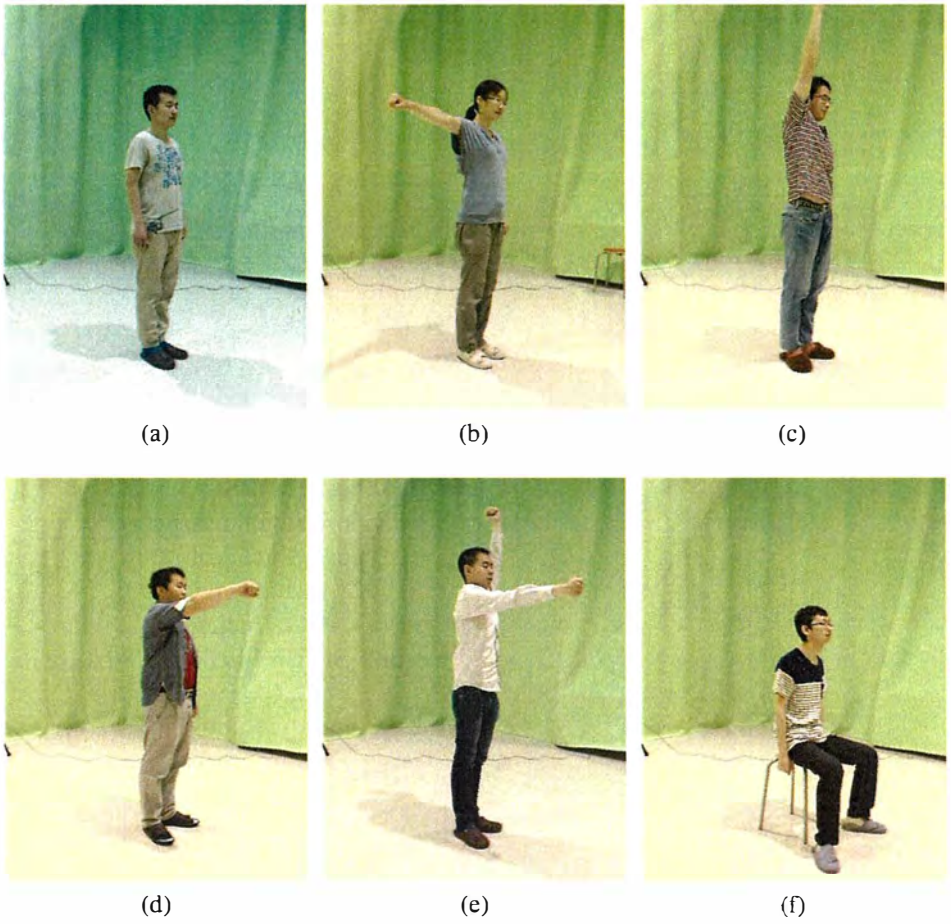
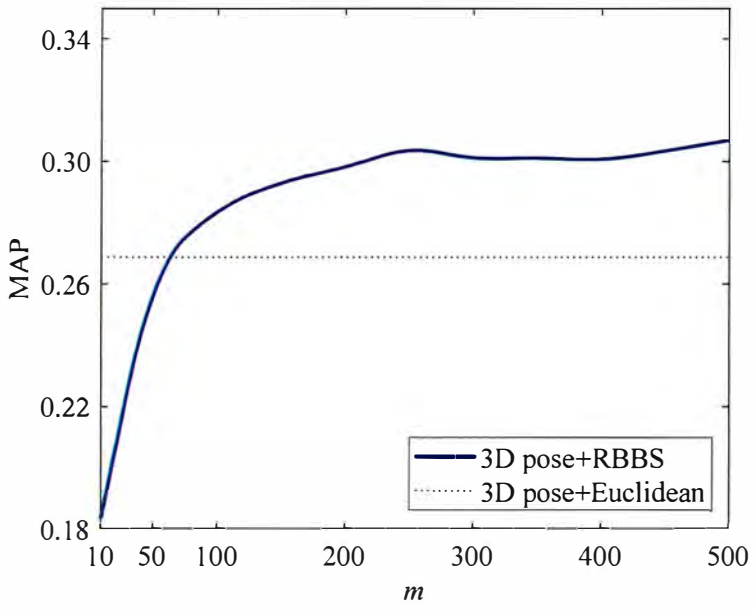
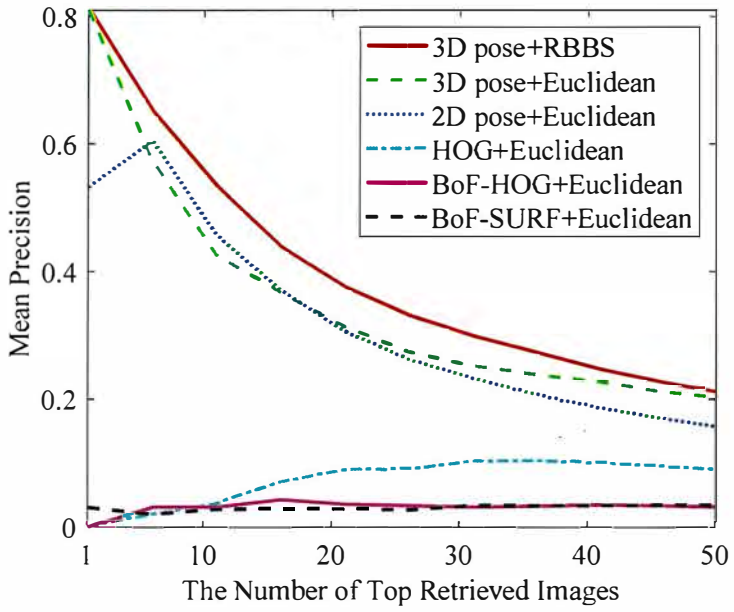


Figure 2.7: Typical examples of dataset images. **(a)~(d)** Poses of stretching arms downwards, sidewards, upwards and forwards respectively. **(e)** The pose that combines left arm pose and right arm pose. **(f)** Pose of sitting straight up with legs naturally bending.

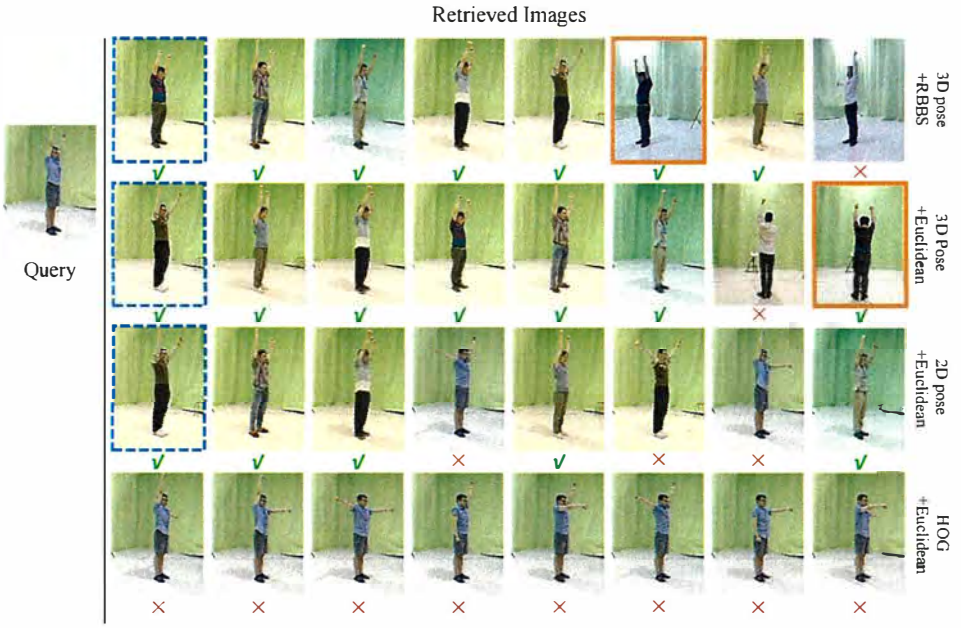


(a)

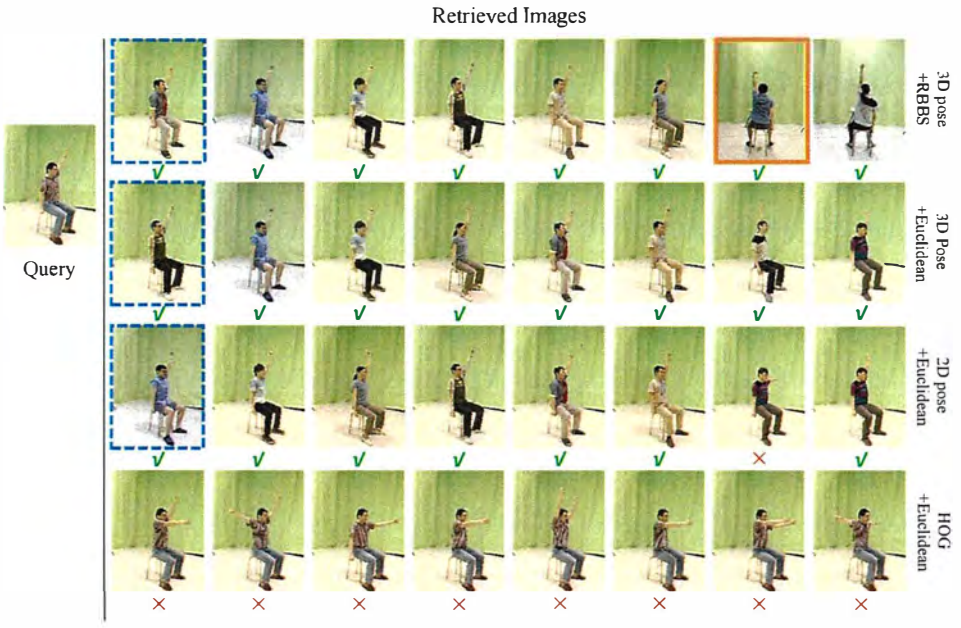


(b)

Figure 2.8: **(a)** Changes of the MAP with the increase of m. The dotted assistant line shows the MAP of 3D pose+Euclidean distance. The MAP increases with the increase of m on the whole. As the increase of m, the MAP is getting smoothing gradually. The conclusion satisfies the synthetic experiments. **(b)** Comparison against existing methods by the precision curve. It indicates that 3D pose+RBBS (proposed method) performs better than other comparative methods. Retrieval by articulated human poses performs much better than traditional image features in VHPR.



(a)



(b)

Figure 2.9: Comparison against existing methods. Retrieved images are arranged from left to right in descending order according to the confidence with respect to different comparative methods. The green ticks below images represent correct matches, while the red crosses represent false matches. The blue dotted boxes highlight the first correct match with the different individual from the query. The orange rectangles highlight the first correct match with different shooting angle from the query.

HOG + Euclidean (Eq. 2.11), BoF+Euclidean, articulated 2D pose+Euclidean, and articulated 3D pose+Euclidean. Referring to the extraction of BoF, I exploit HOG and SURF to extract the feature at key points, denoted as BoF-HOG and BoF-SURF for convenience. 2D and 3D poses are estimated by [?] and [56] respectively. The comparison against image feature-based methods is for revealing their adaptiveness in VHPR.

The quantitative result is plotted in Fig. 2.8(b) by mean precision curve, as well as tabulated in Tab. 2.2 by MAP. Both of the results suggest that methods on the basis of reconstructed 3D poses perform better than traditional image features which are widely used in CBIR. Among the methods with reconstructed poses, the RBBS outperforms Euclidean distance, especially improves the MAP by 14.13% comparing to Euclidean when $m = 500$, and the 3D pose gains higher mean precision than 2D pose. I conclude the discussion according to the observations as 1) As one of the issues, normalization of 2D poses can possibly limit the retrieval performance, causing the lower precision than 3D poses; 2) Although the normalized 3D poses have higher precisions than 2D poses, it remains high reconstruction error and hence the precision of the retrieval with similarities by Euclidean distance is lower than RBBS. conventional feature-based methods are proved to be invalid as all the images are visually similar without considering the pose.

Qualitatively, I show one of the queries and its top-8 retrieve results in Fig. 2.9(a). The first false match of 3D pose+Euclidean distance (ranked 7, row 2) appears anteriorly than that of 3D pose+RBBS (ranked 8, row 1). The correct matches with the different shooting angle from the query of RBBS appear two times, the first of which is two rankings ahead of Euclidean distance (illustrated by orange in Fig. 2.9(a)).

It indicates that the ability to search poses in different shooting angles with respect to the query is weaker than RBBS. On the other hand, the results of 2D poses include more false matches in retrieved images, and all the results are from the same shooting angle. It is worth pointing out that the results of HOG are with the same volunteer in a same shooting angle and appearance, as it is difficult to distinguish poses from visual information by the gradient histogram based feature.

2.7 Conclusion

A challenging retrieval task to retrieve images, which are similar in human pose to the query, is proposed. In order to settle the issues of VHPR, including the definition of "similar poses", normalization, and 3D pose reconstruction error, I propose 3D human pose+RBBS and create an original dataset for evaluation. I reconstruct multiple 3D pose candidates for each image, which are utilized in calculating RBBS. In the experiment, the results indicate that 3D human pose+RBBS performs better than other alternatives. However, one limitation of RBBS is its computation time, because it requires m^2 times computational cost than Euclidean distance.

As future work, accelerating the algorithm by parallelizing the computation of RBBS is considered. Also, as the retrieval is on the basis of the postures, the query in VHPR (human image) can be replaced by a simple drawing. The drawing is often called a sketch, which is discussed in the next chapter in detail.

Chapter 3

Sketch Compression for SBIR by Approximate Representation

[1]This chapter is based on [59]

3.1 Problem Description & Contribution

Conventionally, a sketch is a rapidly freehand drawing that is uncompleted, whereas sketches in SBIR are in a simplified version, which contains only simple strokes and can be described as a binary line drawing. Usually, the application of this kind of sketch is to search relevant images, the objective of SBIR, or 3D models, rather than a constitution of the arts. Although such sketches have been attached by grayscale [60], color [61], psychology (e.g. drawing order) [62], or detailed (e.g. fine-grained) [63] information, the original binary line drawings are most convenient. A binary line drawing is often drawn by the sensor of the drawing board, or touch screen, for it is difficult to be input into the computer from (even scanner cannot input a paper-drawing in binary line drawing), leading to that a general problem of SBIR is that lines are wobbly. To solve the noise of wobbly lines, scholars focus on methods of patch gradient [64] or the main patch gradient [65]. However, despite the uncertain accuracy and long time when compared with complex and large numbers of images, these methods cannot deal with rotation, scaling and translation. The first research that overcomes such hard nuts is by S. Parui et al. [66], who propose to process both sketches and images to segments that include only straight lines. This research achieves not only the improvement of SBIR performance but also the compression of the sketch for SBIR. Nevertheless, only using a straight line to approximately represent a sketch results in some problems, such as a bending curve could be divided into lots of short lines, or it could be far different from the origin.

A sketch is an image apparently. Image compression has been well studied [67,

68], but sketches of SBIR have their specificity, such as binary. If applying normal image compression algorithms to a sketch, not only cannot I apply the compressed sketch to sketch-to-image matching system directly, but they are also slow.

In this chapter, I propose to compress sketch for SBIR by a approximate representation. I extract curves of a sketch, which are segmented. All of the segments are classified into two types, straight lines and arc of circles, with approximately represented. The segment type is more than [66], the segmentation and representation of the proposed method are also different from which. Such representation, which can compress and repaint a sketch approximately, has many applications and advantages. It can be utilized in SBIR and sketch-to-sketch matching, can save much memory of computer. Besides, because the repainting has its own style, this technology can be even associated with the arts. In experiment, I compress sketches in a famous database, named Flickr15k [69] to indicate that the proposed method is fast and can compress sketches by a rather low compression ratio.

3.2 Compression

3.2.1 Preliminaries

A sketch, denoted by $\mathbf{S}(x, y)$, where x and y are coordinates, can be divided into several curves. The curves constituted by points are denoted by $\mathbf{Cr} = \{P_1, P_2, ..., P_n\}$. Each curve can be complex, thus can be further segmented into numbers of segments $\mathbf{Seg} = \{P_1, P_2, ..., P_m\}(m \leq n)$, which include different types. The types in this chapter contain straight line and arc. Finally each segment is represented sparsely as $\mathbf{Rep} = \{\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_k\}$, where $\mathbf{R}$ is the representation of one segment.

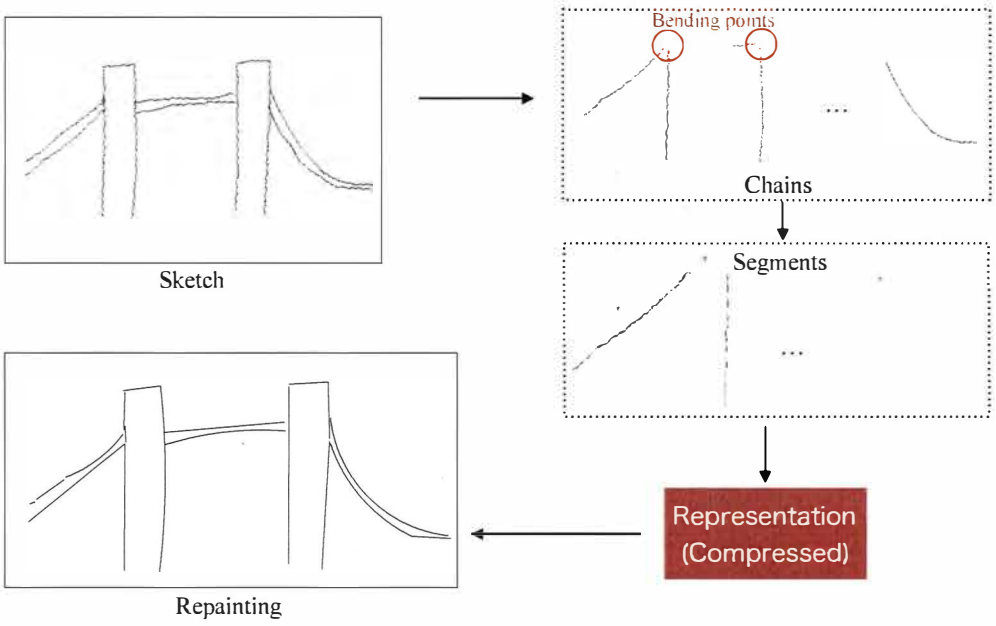


Figure 3.1: Overview of compression. In "Chains", gray points circled by red are bending points for the segment, while the dotted arrows point to the corresponding segments. In "Segments", for straight line, the gray curves are the end-to-end straight line; for arc, the blue curve is constituted by original points and the red curve is regressed based on the blue.

The overview of compression is shown in Fig. 3.1.

3.2.2 Extraction of Chains

The curves that consist of a sketch can be called a chain because it is segmented into multiple segments. Hence multiple chains consist of a sketch. In order to extract each chain of a sketch, an 8-nearest neighbor search (8-NNS) is applied to chain extraction. I visit from the most upper-left black point $\mathbf{S}(x_u l, y_u l)$, in the 3×3 patch center at the location of which, the neighbor black pixel that has not been visited is selected as the next pixel, denoted by $\mathbf{S}(x_n b, y_n b)$ of the chain. Then the next center of the 3×3 patches is located in $\mathbf{S}(x_n b, y_n b)$, which is repeated until there exists not a black neighbor. One of the chains represented by points is extracted finally. The approach to extract **Cr** of a sketch is shown in Algorithm 2, where the center of the being visited patch is denoted by P_{center}, i is the counter of chains and j is a counter of points in a chain.

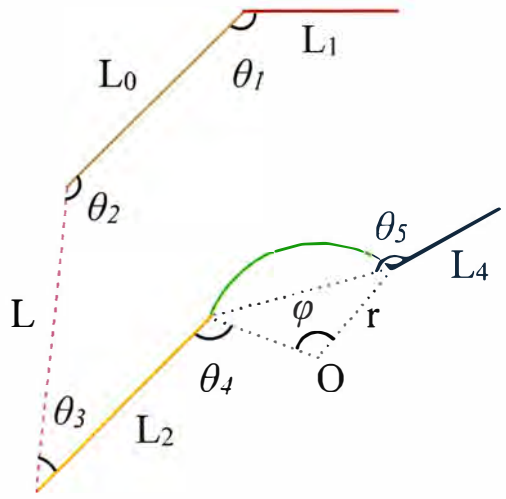
3.2.3 Segmentation

With obtaining chains of a sketch, each of them is segmented into some segments based on curvature. Curvature is usually utilized to evaluate loosely related concepts in geometry. In this case, especially, the curvature is utilized to evaluate bending of black points in a curve. The curvature is computed by

$$K_c = \sum_{i=1}^{d} \omega_i \cdot \angle P_{c-i} P_c P_{c+i}, \tag{3.1}$$

Algorithm 2 Extraction of Chains

Input: $\mathbf{S}(x, y)$
  $i = 1$
  repeat
    $P_{center} = \mathbf{S}(x_u l, y_u l)$
    $j = 1$
    repeat
      Add P_{center} to $\mathbf{Cr}_i$
      $P_{center} = \mathbf{S}(x_{nb}, y_{nb})$
      $j = j + 1$
    until $\mathbf{S}(x_{nb}, y_{nb})$ does not exist
    $i = i + 1$
  until $\mathbf{S}(x_u l, y_u l)$ does not exist
Output: **Cr**



⟨1, L0/L0, 0, 0⟩

⟨1, L1/L0, θ1, 0⟩

⟨1, L/L0, θ2, 0⟩

⟨1, L2/L0, θ3, 0⟩

⟨2, r/L0, θ4, φ⟩

⟨1, L4/L0, θ5, 0⟩

Figure 3.2: Example of representation. Active lines are strokes, while dotted lines are auxiliary lines. The color of strokes corresponds to the representations.

where d is a parameter that denotes the searching degree, c denotes the c th point of the chain, and ω_i is a Gaussian function centered at P_c.

For each chain, c is from $i + 1$ to $l_{cr} - i$, where l_{cr} is the number of points in the chain (length of the chain). After applying Eq. 3.1 to each point, I select bending points by $K_c < \epsilon_1$ (= 2.075). Chains are segmented by such selected bending points, while the segments are represented by sets of points.

3.2.4 Segment Classification

I classify the segments into straight lines and arcs. Because in this research there are only two kinds of curves, I only measure the similarities between original points and end-to-end straight lines. Explicitly, I calculate the mathematical expression of the straight line, $Ax + By + C = 0$, which connects first and last points of a segment, other points on which are utilized to measure similarities with the original points in the curve. The similarity, expressed by distance D, is computed by

$$D = \sum_{i=1}^{m} |PO_i - PM_i|, \tag{3.2}$$

where PO denotes original points and PM is the points given by the mathematical expression.

If $D > \epsilon_2$ (= 1.40), a threshold, the curve is to be regarded as an arc.

3.2.5 Representation

I concentrate on the relationship between a pair of consecutive segments due to the aim that is to cope with rotation, scaling and translation. Further, the first segment

of a sketch is the basic segment, only which can locate the position of every first segment of chains. That is to say, each first segment of chains is paired with the basic segment.

The representation of a segment, paired with the previous segment, can be commonly defined as $\mathbf{R} = \langle type, lr, \theta, \phi \rangle$, as is exampled in Fig. 3.2. In the representation, lr is length ratio computed by

$$lr = \begin{cases} \dfrac{l_c}{L_0}, & \text{basic segment is a straight line} \\ \dfrac{l_c}{r_0}, & \text{basic segment is an arc} \end{cases} \tag{3.3}$$

In Eq. 3.3, basic segment is the first segment of the first extracted chain, if the first segment is a straight line. Let L_0 be the length of the basic segment, or r_0 be the radius of the basic segment, and l_c be the length of current line or radius of current arc.

Actually the representations of straight lines and arcs are different, such as the definition of θ and ϕ, respectively stated in Section 3.2.5 and 3.2.5. The θ of basic segment is 0.

Straight Line

Representation of a straight line can be especially defined as $\mathbf{R}_{sl} = \langle 1, lr, \theta, 0 \rangle$. In case that $D <= \epsilon_2$, I consider the segment can be regarded as a straight line that is connected end-to-end from the first points of the segment to the last. In order to represent the segment compared with the paired segment, if the paired segment is a straight line, θ is set as the minor angle from the previous to itself, while if the

paired segment is a sector, θ is set as the minor angle from an auxiliary line, which connects the first and last points on the arc, to itself. The ϕ of a straight line is always 0.

Arc

Representation of an arc can be especially defined as $\mathbf{R}_{st} = \langle 2, lr, \theta, \phi \rangle$. In case that $D > \epsilon_2$, we recognize the segment as an arc. By the function

$$\{O, r, \phi\} = \text{Regression}(\mathbf{Seg}), \tag{3.4}$$

I can obtain the information of the regressed arc, including center (O), radius (r) and center angle (ϕ), which is the same to the ϕ in the representation of an arc.

To compute the ϕ of an arc, it is necessary to draw the auxiliary line, too, which is from the first point of the sector to the center. This auxiliary line replaces the position of the end-to-end straight line in Section 3.2.5 to compare with the paired segment.

3.3 Decompression

Decompression is to repaint the sketch in fact, the algorithm of which is shown in Algorithm 3. After setting the start angle ϕ_{start} and start point P_{start} for the representation of the basic segment, each segment is visit for repainting. The function P =Next($\cdot$) denotes computing the ending of the segment, while the function $Paint(\cdot)$ denotes painting the segment being visited.

Algorithm 3 Decompression

Input: **Rep**, ϕ_{start}, P_{start}
  Set ϕ_{start} and P_{start} for $\mathbf{R}_{1,1}$
  $P_{\text{next}} = \text{Next}(\mathbf{R}_{i,j}, P_{\text{start}})$
  $\text{Paint}(\mathbf{R}_{1,1}, P_{\text{start}}, P_{\text{next}})$
  $i = 1$
  repeat
    $j = 1$
    repeat
      $P_{\text{next}} = \text{Next}(\mathbf{R}_{i,j}, P_{\text{current}})$
      $\text{Paint}(\mathbf{R}_{i,j}, P_{\text{current}}, P_{\text{next}})$
      $j = j + 1$
    until Each segment of the chain has been visited
    $i = i + 1$
  until All the chains have been visited
Output: $\mathbf{S}(x, y)$

3.4 Experiment

3.4.1 Numerical Evaluation

I compress query sketches in Flickr15k [69] for numerical evaluation, which is shown in Tab. 3.1. The first column represents the average compression ratio of the 330 sketches in Flickr15k, which is computed by

$$\text{Compression Ratio} = \frac{\text{Compressed Size}}{\text{Uncompressed Size}}. \tag{3.5}$$

For evaluation, a lower compression ratio means a more strong compression. The compression ratio of the proposed method is rather low, indicating that the compression of the proposed method is strongly effective.

The second and third column represents the compression and decompression speed per sketch. They indicate the speed is acceptable but far from real-time. Moreover, the speed of decompression is much lower than compression by reason

that decompression includes a procedure to generate a normal sketch.

3.4.2 Visual Experiment

Some of the visual results are shown, which are the liner approximately represented sketches and contour image, as is shown in Fig. 3.3. The original sketches are some of the sketches in Flickr15k, I compress and decompress them to repaint them. And contour of the image is extracted for compression and decompression. Such visual results indicate the compression can preserve the character of the original sketches.

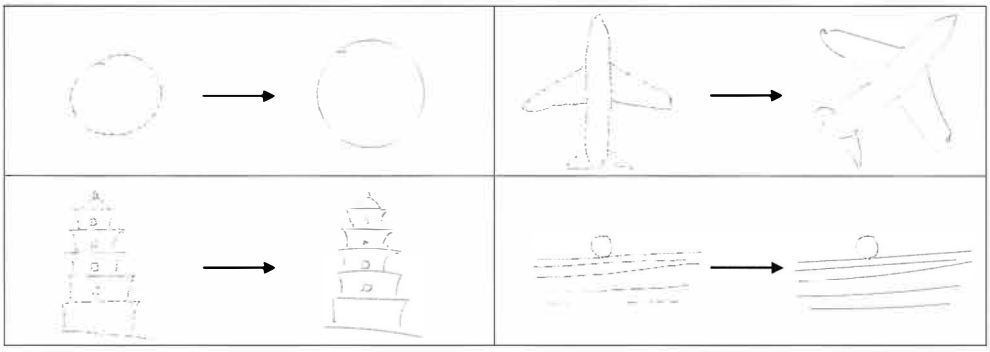
3.5 Conclusion

A sketch compression by approximate representation is proposed in this chapter, which is designed for SBIR but not only limited to SBIR. The character of sketches to divide sketches into chains and further into segments is exploited, which can finally be represented in a low dimension. In the experiment, 330 sketches from Flickr15k and some contour images converted from natural images are compressed. Both the qualitative and quantitative results indicate that the proposed compression method has a striking effect. On the other hand, the method can not realize real-time processing at the current time, which is a work I plan to focus on in the future.

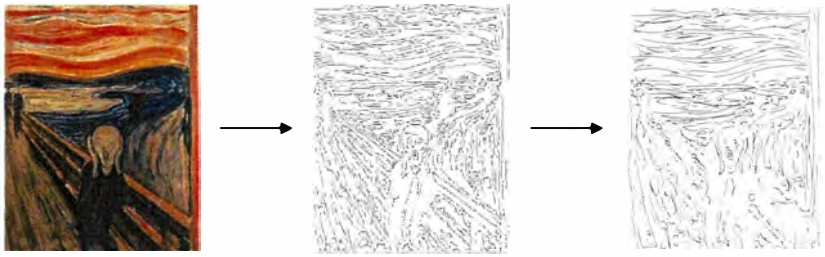
VHPR and SBIR are two IR research fields with various queries. In order to further systemize the queries and find the commonalities of different queries between IR and OD, the next chapter discusses a category of queries (the product image) for OD.

Table 3.1: Numerical Evaluation

Compression Ratio	Compression Speed (s)	Decompression Speed (s)
0.023%	0.3231	0.7497



(a) Compression from sketches



(b) Compression from contour image

Figure 3.3: Visual results. (a) Sketch on the left of arrows are uncompressed (original) sketches, sketches on the right of arrows are repainted according to compressed data. (b) From left to right, color image, contour image, compressed sketch representation of the proposed method.

Chapter 4

TemplateFree: Product Detection on Retail Store Shelves

[1]This chapter has been published in [70]

4.1 Problem Description & Contributions

Product arrangement for retail store shelves is important and currently inspected by clerks manually thus costly. On one hand, products under contracts should follow the compliant arrangement according to the contracts. They often appoint the number and position of product sets on a store shelf, which can further affect customer satisfaction [71] and sales performance [72]. On the other hand, manual inspection and management not only spend human resource but can also disturb shopping customers. Clerks must inspect and manage all of the shelves several times every day, but shelves can be disarranged within a short time after the management. Such issues push the appearance of automatic product recognition systems. Among the alternative data categories, such as depth images and multi-view images, the single shelf RGB images require the most convenient device (monocular camera).

However, the single RGB images based product recognition faces many challenges. Firstly, the recognition usually turns out to be a one-shot issue, for it costs a lot to collect the product images for machine learning in multifarious environments. Hence the training data is template product images actually. Secondly, the illumination of each shelf is diverse. Thirdly, some packaged products (e.g. bagged potato chips and laundry detergent refills) can be distorted. Fourthly, the background involves varied noise. The noise can include the advertisement of a product, which is visually more similar to the template product image than the corresponding real product on shelves. The above challenges make it difficult to compare a template product image with the real product images. Lastly, the difference between products in the same brand can be rather small.

(a)

(b)

Figure 4.1: The defects of product recognition with templates. (a) Examples of the retail store shelf and detected products illustrated by green by the proposed method. (b) The corresponding postured product image of products in (a). The corresponding products in (a) are visually different from (b) caused by illumination, noise, and distortion.

Further, the research field on product recognition is still in its infancy. Related works apply the workflow of multi-objects detection and recognition to product recognition for the solution of the above challenges. The workflow requires the templates to satisfy the multi-objects detection algorithms, which are used to find similar regions in the store shelf images (target images). The most simple ideas (e.g. [73, 74]) treat the templates as the training data for the classifier of product recognition. The performances of these methods rely on the performance of the corresponding object detection algorithm very much. Although video data [75, 76] and additional information (e.g. words on products [77] and planogram [78]) is then leveraged to product recognition, the scarcity of related journal articles and patents, as well as the inadequacy of product species in experiments, reveal that product recognition is at a prototype stage.

The application of template product images to locating products has numerous defects. There exist two choices for the templates, i.e., master images and postured product images. The master image is the blueprint of the product appearance, while to obtain the postured product image that is exampled in Fig. 4.1(b), it is necessary to place the product in a simple background and take the photo. Postured product images are extraordinarily difficult to be collected and updated. By comparison, master images are easier to be collected and updated, but implicate copyright issues and are more difficult for recognition owing to the more visual difference caused in production. Moreover, unless gathering all categories of the template product images, both choices probably involve miss-detection.

In order to minimize the defects of using templates, the best approach is to detect products without templates by the zero-shot learning method [79], then recognize

each detected product. Zero-shot learning can recognize an unseen class that is not labeled and trained directly but can be inferred from the seen classes (labeled in the training data). Zero-shot learning is an extreme form of transfer learning [80]. Especially, the zero-shot object detection (ZSD) methods require semantic label space and build the bridge between seen and unseen classes by introducing semantic similarity embedding [22, 23].

For the solution of all the above issues, in this chapter, TemplateFree, a zero-shot deep learning based product detection method by single retail store shelf images, is proposed. Instead of the common idea that detecting and recognizing products according to the templates, TemplateFree avoids templates and concentrates on the characteristic of retail store shelves that each shelf can be segmented horizontally into several layers and vertically into products. Both horizontal and vertical separatrix candidates are sampled and optimized, then the vertical separatrices are refined. The optimization and refinement of vertical separatrices are assisted by the trained GoogLeNet [81]. The GoogLeNet learns whether a region is a single product through zero-shot learning. In addition, I consider the recognition can be on the basis of the detection, regarding the detection as the first step and preliminary means. To the contrary, shrinking the *region of interest* (ROI) by detection improves the performance of recognition. TemplateFree works well even via zero-shot learning (i.e., the classifiers are learned by completely irrelevant training data).

4.2 Related Work

The attempt of convenience for the management of product arrangement starts with semi-automatic product recognition. Nowadays, the manual management relies on not only the visual inspection but also the recognition by bar code scanner [82]. Tsai et al. [73] propose a remote visual mobile product recognition method, which requests users take the photo of each product by their mobiles and recognizes products by the remote server. The operation of the method for clerks is only a little more convenient than the bar code recognition but sacrifices the promising precision of bar code. Winlock et al. [75] propose a video-based real-time product recognition system, named ShelfScanner. ShelfScanner recommends users to scan products in the store shelf one by one, with building the structure of the shelf. ShelfScanner is the first system that can recover whole shelves. Lopez et al. [83] propose another novel assisted shopping system for blind people by *radio frequency identification devices* (RFID) and QR-code, whereas the method works through additional devices. Kassim et al. [84] propose MyHalal, a system for recognizing whether a product is a Halal product or not, realized by the smartphone camera and bar code reader, but with limited scope. All in all, in spite of the more or less improvements comparing to the bar code scanner, the above researches still need the manual operation.

As is difficult to collect the training data, fully automatic product recognition is usually a few-shot or even one-shot matching problem. Although non-learning matching by manual features methods, e.g. *histogram of oriented gradients* (HOG) [2] and *speeded up robust features* (SURF) [1], have a great performance once, they are

outperformed by deep learning methods gradually with the improvement of image recognition and object detection. The methods for object detection, such as YOLO [85, 28, 29] and R-CNN [86, 87, 88], provide the basis for fully automatic product recognition.

The fully automatic product recognition system analyses the whole shelf images, which requires only a fixed monocular camera for a shelf. George et al. [89] present a per-exemplar multi-label image classification method and create a large product dataset, named Grocery Products dataset. Marder et al. [76] propose to monitor retail store shelves by image analysis, which apply the state-of-the-art object detection method at that time. Varol et al.[90] decompose the problem of product recognition into detection and recognition clearly, which are solved by a generic product detection module [91] and *support vector machines* (SVM) [92] respectively. George et al. [77] first introduce the characteristic of products that brands printed on the products often contain texts. They classify the products into brand-level classes by text recognition and active learning based classification. Their method cannot classify the products into specific product-level classes and the text recognition can be useless once the brand contains no text. Tonini et al.[78] add the planogram as an additional condition and propose to recognize products as a sub-graph isomorphism problem. Notwithstanding their method achieves a rather excellent result, as the planogram changes constantly, it is difficult to apply. Geng et al. [74] use feature-based matching and one-shot deep learning to conduct a coarse-to-fine product detection and recognition. With the robust ability of deep learning to understand details, their method can distinguish different products that have small differences. However, the above typical present fully automatic methods

treat product recognition as one-stage research of multi-objects detection problem, which requires templates.

In order to abandon the templates, a zero-shot method is necessary. Zero-shot learning is first proposed by Palatucci et al. [79] for image clustering, aiming to predict unseen classes that are not labeled in training data for clustering. Gavves et al. [93] propose a zero-shot active learning that reuses and revisits the old datasets, whereas their method needs a human to teach and annotate the names of the new classes. Kodirov et al. [94] propose to add the constraint that the coded data must be able to reconstruct the original visual feature. One common point of the above zero-shot learning can be concluded as they learn a projection from the training data (seen classes) to the semantic embedding space. Inspired by this, zero-shot learning is applied to object recognition [95] and detection [23, 22]. Demirel et al. [23] propose a hybrid region embedding for ZSD that combine two mainstream embedding approaches in zero-shot learning. Bansal et al. [22] propose the first visual-semantic embedding for ZSD. Nevertheless, indiscriminately imitated ZSD methods to product detection is difficult because it can omit many of the neatly arranged products in shelves by the present sampling methods.

In this chapter, a zero-shot deep learning based product detection method, TemplateFree is proposed, which abandons the templates and locate the products by segmenting shelves horizontally and vertically. In TemplateFree, the sampling depends on the segmentation, reducing the omission of products.

4.3 TemplateFree

4.3.1 Overview

TemplateFree can be decomposed into four parts, as shown in Fig. 4.2, horizontal segmentation, layer classification, vertical segmentation, and refinement. Horizontal segmentation can segment a whole store shelf into several layers (see Fig. 4.2(b)). Each layer exhibits a row of products (product layer) or a non-product layer, classified by a trained GoogLeNet. Within every product layer, the vertical separatrix candidates are detected (see Fig. 4.2(d)), which are the foundation for segmenting the product layer vertically into products. The vertical segmentation is assisted by another trained GoogLeNet. Last, the border of products and the worst separatrices are refined, the effect of which is exampled by the comparison between Fig. 4.2(e) and (f). The four parts are introduced in Sect. 4.3.2~4.3.5 specifically and respectively.

4.3.2 Horizontal Segmentation

The horizontal segmentation consists of horizontal separatrix candidates detection and the optimization of candidate combinations, displayed in Fig. 4.3.

Horizontal Separatrix Candidate Detection

In retail store shelf images, there exist several kinds of horizontal bar-like objects that can segment the shelf into several (product or non-product) layers, such as clapboards and the top of some categories of products. Therefore, the horizontal separatrix candidates detection is to locate the horizontal bar-like objects. I con-

sider the bar-like objects locating is analogous to various problems such as seat-belt detection. Referring to the non-learning method for seat-belt detection proposed by Guo et al. [96], I detect the horizontal separatrix candidates by preprocessing and Hough transform. The shelf images are preprocessed via grayscale converting, Gaussian blur, horizontal Sobel edge detection, and dilation. The grayscale converting and Gaussian blur can filter much clutter, horizontal Sobel edge detection minimum the vertical noise, and the dilation makes the remained dominant pixels (non-black pixels) more robust, which is advantageous to Hough transform based straight line detection. The detected straight lines stretch to many orientations. They are filtered according to the distance in height between the endpoints. Although the filtering guarantees the remained straight lines are regular in orientation, it is difficult to guarantee an adaptive number of the straight lines because of the diverse and complex illumination and saturation. Thus I recur the Hough transform with increasing the parameter of Hough transform that restricts the minimum length of the detected lines (horizontal separatrix candidates) until the number of detected lines is within a predetermined range or exceeding a predetermined loop number.

Horizontal Separatrix Candidate Optimization

Despite the limited quantity and orientation, the candidates can be in an uneven distribution or contain too short instances. The problem can invoke the misdetection. I solve the problem by optimizing the combined candidates. The objective function of the optimization can be written as

$$\min E_l + E_h \,, \tag{4.1}$$

where E_l and E_h denote the evaluation of the candidate in lengths and candidate spacing heights respectively. The effect of E_l constraints the lengths so that too short candidates are identified as the inappropriate, while E_h helps to regularize the candidates. I calculate E_l by

$$E_l = \frac{1}{\text{mean } \mathbf{L}}, \tag{4.2}$$

where $\mathbf{L}$ denotes the set of each candidate's lengths. The calculation of E_h is

$$E_h = \ln\left(e + \frac{1}{\min \mathbf{H}}\right) + \min\left\{\mathrm{D}(\mathbf{H}), \mathrm{D}(\mathbf{H}')\right\}, \tag{4.3}$$

where $\mathbf{H}$ denotes the set of candidate spacing heights, $\mathbf{H}'$ denotes the set of spacing heights of candidates incorporating the top and bottom of the image, and $D(\cdot)$ denotes the variance. In Eq. 4.3, the first term prevents the case that all of the combined candidates appear within a concentrated region, while the second term constrains the combined candidates are in a uniform distribution by measuring the variance of the spacing heights. Because the computational cost of Eq. 4.1 is low, I optimize via sampling all the possibilities, i.e., I optimize $\sum_{i=3}^{k} \mathrm{C}_k^i$ categories of candidate combinations, where $k \in \mathbb{Z}$ is the number of the straight lines.

4.3.3 Layer Classification

Some of the layers segmented from shelf images may contain no products, named non-product layers. Contrarily, the layers that contain products are named product layers. It is useless to detect products in non-product layers, hence the layers are

classified by a zero-shot learned GoogLeNet, named layer GoogLeNet (L-GoogLe).

GoogLeNet

GoogLeNet is a convolutional neural network for image recognition, which is expanded to deeper than 100 layers but maintains the computational budget constant. I select GoogLeNet because of its high precision and its utilization and application as the backbone framework in many later pieces of research. More details of GoogLeNet can be found in [81].

Zero-shot L-GoogLe

In this section, I intend to recognize unknown shelf layers (unseen class, T_l) according to known shelf layer images (seen classes). The seen and unseen classes are defined to be completely visually different in the shelf category, product category, and background. They also can be different in illumination or slightly different in shooting angle. An shelf layer image can be divided into product layers (K_l), non-product layers (K_n), and background (B). Common learning model can be concluded as $(K_l, K_n) \rightarrow K'_l$, where K'_l is the target to be predicted in the same class to K_l. However, T_l is much visually different to K_l, being considered in different classes. I thus embed the simple semantics that

- "the product layer is different from the background",

in order to take the advantage of B, and conduct $(K_l, K_n, B) \rightarrow T_l$, where K_l can be regarded as the positive training data, while K_n and B can be regarded as the negative training data.

It is necessary to prepare K_l, K_n, and B. The original training data is the store shelf image. I segment the images by the method in Sect. 4.3.2 and the annotated layer images compose K_l and K_n. Then I uniformly randomly sample background image patches in the images as B. All of K_l, K_n, and B are resized to 224×224 pixels before training. The positives and negatives are exampled in Fig. 4.4(c)~(d).

It is worth pointing out that L-GoogLe can be regarded as the zero-shot learning framework if and only if it satisfies the assumption that T_l is much visually different from K_l (e.g. Fig. 4.4(a) and (c)). Otherwise, T_l can be predicted by $(K_l, K_n) \rightarrow T_l$.

4.3.4 Vertical Segmentation

By means of the characteristic that the products in a layer can be segmented into single instances by the shadow between each pair of products, the problem can be converted to shadow detection. Products are in diverse categories, various in shape, illumination, and scale, resulting in the difficulty of direct detection. On the contrary, the shadow is often the comparatively dark area of the layer image and near to black, the detection of which can be much easier and more precision. I thus detect the vertical dark areas in the layer image as the vertical separatrix candidates. However, some of the dark areas are not the shadow between adjacent products, I group the detected candidates and optimize the grouping.

The workflow of vertical segmentation is shown in Fig. 4.5, including vertical separatrix candidates detection and grouping.

Vertical Separatrix Candidate Detection

The candidates are detected on the basis of preprocessing. A layer image is preprocessed by grayscale converting, Gaussian blur, vertical Sobel edge detection, binary Otsu thresholding [97], and horizontal erosion. Different from the preprocessing in Sect. 4.3.2, which utilizes dilation to enhance the straight line detection, the horizontal erosion replaces the dilation to further reduce the noise. W.r.t. Otsu thresholding, I apply it for decreasing the sporadic noise and weak gradients due to its low computational cost.

The pixels of preprocessed layer images are binary, including dominant pixels and featureless pixels, as the respective white and black pixels of the preprocessed images in Fig. 4.5(b)~(d). Since the layer images can include noise in the top and the bottom, the dark areas are more precision to be detected by shrinking the ROI of layer images lengthwise. Hence, I measure and linearly minimize the density of dominant pixels to shrink the ROI lengthwise. The density ρ is calculated by

$$\rho = \frac{n}{A}, \tag{4.4}$$

where n denotes the number of dominant pixels and A is the area.

I search the ROI of preprocessed layer images from left to right by a predetermined stride and window width, with counting the number of dominant pixels in each window. The numbers of dominant pixels are utilized to create the histogram, in which the local maxima are treated as the separatrix candidates, as is illustrated by the yellow dotted lines in Fig. 4.5(b)~(d).

Algorithm 4 Vertical Separatrix Candidates Grouping

Input: $\mathbf{X}$, $\mathbf{V}$, w_{max}
  $\delta = w_{max} \times 0.6$, $j = 1$
  repeat
    $b = 1$, $t = 2$
    $\mathbf{S}$.append(b)
    repeat
      if $\mathbf{X}_t - \mathbf{X}_b \geq \delta$ **then**
        a = select($\mathbf{V}, b, t$)
        $\mathbf{S}$.append(a)
        repeat
          $b = b + 1$
        until $\mathbf{X}_b - \mathbf{X}_a \geq \delta \times 0.7$
        $t = b + 1$
      end if
    until $\neg$ exist($\mathbf{X}_b$)
    $\delta = \delta + w_{max} \times 0.2$
  until $\delta > w_{max}$
  $\mathbf{S}_{opt} = \min \forall f(\mathbf{S})$
Output: $\mathbf{S}_{opt}$

Vertical Separatrix Candidate Grouping

Because the candidates suffer error detection, probably caused by the residual noise. The issue is solved by grouping. Instead of the common grouping that groups the candidates by an experienced width δ, I handle δ as a variable, in order to optimize the combination of the candidates. The algorithm for grouping in the shrunken ROI is described in Algorithm 4, where $\mathbf{X}$ denotes the set of candidates, $\mathbf{V}$ denotes the set of dominant pixel quantities corresponding to each candidate, w_{max} is the maximal distance between each pair of candidates, $\mathbf{S}$ denotes the set of grouped candidates, and $\mathbf{S}_{opt}$ denotes the optimization of $\mathbf{S}$. The function select($\mathbf{V}, b, t$) is to select the maximum $\mathbf{V}_a$ between $\mathbf{V}_b$ and $\mathbf{V}_t$, where a is the index of the maximum and used as the return. The reason that I select the region that contains the largest number of dominant pixels is that more dominant pixels represent a candidate that more

probably turns out to be a separatrix, whereas false candidates usually contain the low quantity of dominant pixels. The function f is calculated by

$$f = \min \frac{\text{mean S-GoogLe}(\mathbf{P})}{\text{mean S-GoogLe}(\mathbf{N})}, \tag{4.5}$$

where **P** and **N** are the potential positive and negative regions, corresponding to the $\mathbf{N}_{1\sim2}$ (illustrated by red) and $\mathbf{P}_{1\sim5}$ (illustrated by green) in Fig. 4.5(e). Specifically, a layer image can be segmented by **S** into several regions, of which the leftmost and rightmost regions are potential negative regions **N**, while the other regions are potential positive regions **P**. Each potential positive region is optimized to be recognized as the region that contains a single product, while the potential negative regions are optimized to be recognized as non-product regions, partial product regions, or multi-products regions. Furthermore, the potential negative regions are selected according to the widths of the adjacent end positive regions (e.g. $\mathbf{N}_1$ is selected according to $\mathbf{P}_1$ in Fig. 4.5(e)). The widths of the potential negative region and the end region of potential positive regions are aequilate, if the width is less than the width between the corresponding end separatrix and the border of the layer image. Otherwise, the potential negative regions are selected between the end separatrix and the border of the layer image.

The function S-GoogLe(·) is based on another zero-shot trained GoogLeNet, named single-product GoogLeNet (S-GoogLe) specifically described in Sect. 4.3.4. Besides, I repeat the vertical separatrix candidate detection from the Gaussian blur in preprocessing and update the candidates, in order to adapt the range of the grouping stride.

Zero-shot S-GoogLe

I denote the p classes of training product data by $K_s = \{C_1, ..., C_p\}$, then $(K_s, B) \rightarrow (C'_1, ..., C'_p)$, where $(C'_1, ..., C'_p)$ are the products in the same classes to K_s respectively. The purpose of S-GoogLe is to predict a new class of product, C_q, which is not a part of K_s. I embed the simple semantics that

- "a single product is different from a partial product",
- "a single product is different from a multi-products".

Hence I conduct $(K_s, K_h, K_m, B) \rightarrow C_q$ as a zero-shot learning model, where K_h denotes partial images extracted from K_s and K_m denotes multi-products images from the original training data. In this model, K_s is regarded as the positive data, while K_h, K_m, and B are regarded as the negative data, as is shown in Fig. 4.5(g)(h).

4.3.5 Refinement

The proposed grouping searches from left to right, possibly leading to the case that the leftmost positive region is not a product region, as well as the case that ground-truth borders are possibly not detected as the candidates owing to its visual difference from the shadow between adjacent products. Also, the grouping limits each distance between continuous vertical separatrices within δ, though δ is adjustable. Consequently, I propose to refine the detected vertical separatrices.

The refinement is decomposed into two steps, border refinement, and worst interiors refinement. I attempt to insert or delete vertical separatrix candidates that are not treated as the detected separatrix around both borders. All the combinations of the candidates between the border of the layer image and the second nearest

separatrix to the border are evaluated by means of Eq. 4.5, the optimal combination persists. For the worst interiors refinement, which is evaluated as the worst by max S – GoogLe(**P**), I refine within the worst region and its neighbor regions. The region is denoted by R_l. Instead of using global Eq. 4.5, i.e., optimizing the whole layer image, I optimize only the local region R_l to decrease the computational cost. Similar to the border refinement, I evaluate all the possible combinations of unused local candidates, then the optimal combination is assembled with other separatrices. The assembled combination is evaluated globally. If the global evaluation is better than the former combination, it replaces the former one.

4.4 Experiment

The dataset I utilize is introduced in Sect. 4.4.1. The protocol for evaluating the proposed method is introduced in Sect. 4.4.2. I evaluate the effect of relevant training data quantity in Sect. 4.4.3, and evaluate the refinement in Sect. 4.4.4. I also conduct the comparative experiment to the robust method, faster R-CNN [88], in Sect. 4.4.5.

4.4.1 Dataset

I use Grocery Dataset [90] and Grocery Products dataset [89]. Grocery Dataset contains 354 grocery store shelf images, involving 10 brands, which are collected from 40 groceries with 4 cameras. Grocery Products dataset contains 680 test images and 8350 training images including 80 categories of products. However, the images in Grocery Products dataset are not the whole shelf images, instead, they are the

partial shelf images. Such dataset cannot be used to evaluate the proposed method. Therefore, Grocery Products dataset is used to train the GoogLeNets only. That the categories of the two datasets are independent of each other, as is exampled in Fig. 4.6, guarantees the experiments are zero-shot learning.

4.4.2 Evaluation Protocol

I evaluate L-GoogLe by success rate, as well as evaluate the performance of TemplateFree and the comparable method by the recall, precision, and f-measure. I determine a detected instance (instances in this research are the products) is correct if the overlap coefficient is more than 0.5. The recall is the fraction of the number of correctly detected instances over the number of ground truth. Precision is the fraction of the number of correctly detected instances among the number of totally detected instances. Recall and precision are unilateral. E.g., extreme precision can be 1 and unreasonable when a method detects only one correct product but the image contains many products that should be detected, recall can also be 1 in the situation that a method detects all the products but meanwhile detects large numbers of false instances. Accordingly, F-measure is utilized. F-measure is the harmonic mean of recall and precision, calculated by

$$F = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{4.6}$$

4.4.3 Effect of the Training Data

In order to evaluate the effect of the training data, 70% of Grocery Dataset is utilized for test, while 0 ~ 30% of Grocery Dataset along with Grocery Products dataset are utilized for training. When the training data contains 0% of Grocery Dataset, the learning is pure zero-shot learning.

The change of L-GoogLe is displayed in Fig. 4.7(a). As the quantity of Grocery Dataset in training data increases, the success rate is approximate to 1 continuously and changes little. The lowest success rate is 97.85% given by pure zero-shot learning. It indicates the correlation between training data and test data affects the performance of L-GoogLe little. The zero-shot trained L-GoogLe achieves a rather high success rate to promise the performance of layer classification. Further, it is barely necessary to update the training data and retrain, which reduces lots of costs in the real application.

The change of S-GoogLe is displayed in Fig. 4.7(b). With the increasing quantity of the relevant data for training, the precision changes little, while the recall and the F-measure increase overall. The little change and high value of the precision separately reveal the increment of relevant data affects the precision little, and once a region is detected by the proposed method, it has a high possibility to be a product region. The recall is not as high as the precision, which indicates some products are not detected. Aside from the failed grouping (described in Sect. 4.3.4) and refinement (described in Sect. 4.3.5), it can also be mattered by the undetected or falsely detected horizontal separatrices, false classification of layers, and undetected or falsely detected vertical separatrices. Although the fail of each step decreases the

performance, it still achieves a feasible recall even with the pure zero-shot learning. Referring to the effect of the training data, the recall is affected slightly. The recall increases slowly with the increasing quantity of the relevant training data. The F-measure also increases with the quantity of the relevant training data ascribed to the increase of the recall. As a conclusion, the proposed method performs well even with pure zero-shot learning but can perform better via increasing the relevant training data.

4.4.4 Effect of Refinement

I compare the refined results to the unrefined results numerically in Fig. 4.7(c). Comparing to the unrefined results, all of the precision, recall, and F-measure is improved by the refinement. The recall has been improved most. The worst regions refinement improves each region more probable to a single product. The border can be extended so that the product region that is close to but excluded by the vertical segmentation can be detected. The precision has been improved as well. The leftmost border can be false detection because the leftmost vertical separatrix candidate is always used, so the border refinement effectively reduces the appearance of such false detection. The F-measure has also been improved along with the improvement of the precision and recall. The results indicate the refinement is effective.

4.4.5 Comparative Experiment

I compare the proposed method with faster R-CNN [88]. Both of the training of the proposed method (including training L-GoogLe and S-GoogLe) and faster R-CNN

are trained by the pure zero-shot dataset. The comparison results are demonstrated in Fig. 4.7(d), where TemplateFree leads in the comparison of recall, while the precision of faster R-CNN is a bit higher. The higher precision of faster R-CNN indicates false detection by faster R-CNN is less than TemplateFree. However, the comparison by the recall indicates that faster R-CNN detects only 11.95% of the ground truth, much less than the 67.79% by TemplateFree. The F-measure of faster R-CNN is far less than TemplateFree, meaning comprehensively TemplateFree performs better than faster R-CNN.

I example in Fig. 4.8 the visual comparison between TemplateFree and faster R-CNN. Faster R-CNN can detect only a few products, which can be effected by the different classification and resolution from the training data. On the other hand, TemplateFree reduces such influences and performs much better. However, in Fig. 4.8(b), there still exist false detections and miss detection, which are typical for TemplateFree. TemplateFree cannot filter the non-product regions that are between products, such as the false detection in the first layer and the left four false detections in the second layer. The second false detection from right in the second layer is caused by the vertical separatrix candidate detection, which fails to detect the separatrix candidate between the pair of products. The right-most false detection in the second layer and the miss detection in the third layer are also caused by the miss and false separatrix candidate detection. Although they may be revised by iterative refinement, whereas the operation is not conducted because the iterative refinement leads to the expensive computational cost.

4.5 Conclusion

A method of zero-shot deep learning based product detection on retail store shelves, named TemplateFree is proposed. TemplateFree works without template images and has four steps, including horizontal segmentation, layer classification, vertical segmentation, and refinement. The layer classification is on the basis of L-GoogLe, as well as the vertical segmentation and the refinement depends on the S-GoogLe. In experiments, I evaluate TemplateFree quantitatively, which indicates the proposed method performs better than the existing alternative method. For future work, as TemplateFree is not robust in the situation that a region between products is empty, I deem the feedback from product recognition can improve the performance. Also, instead of learning, the separatrix candidate detection is experienced, causing many false and miss detection. So I intend to improve the performance of separatrix candidate detection.

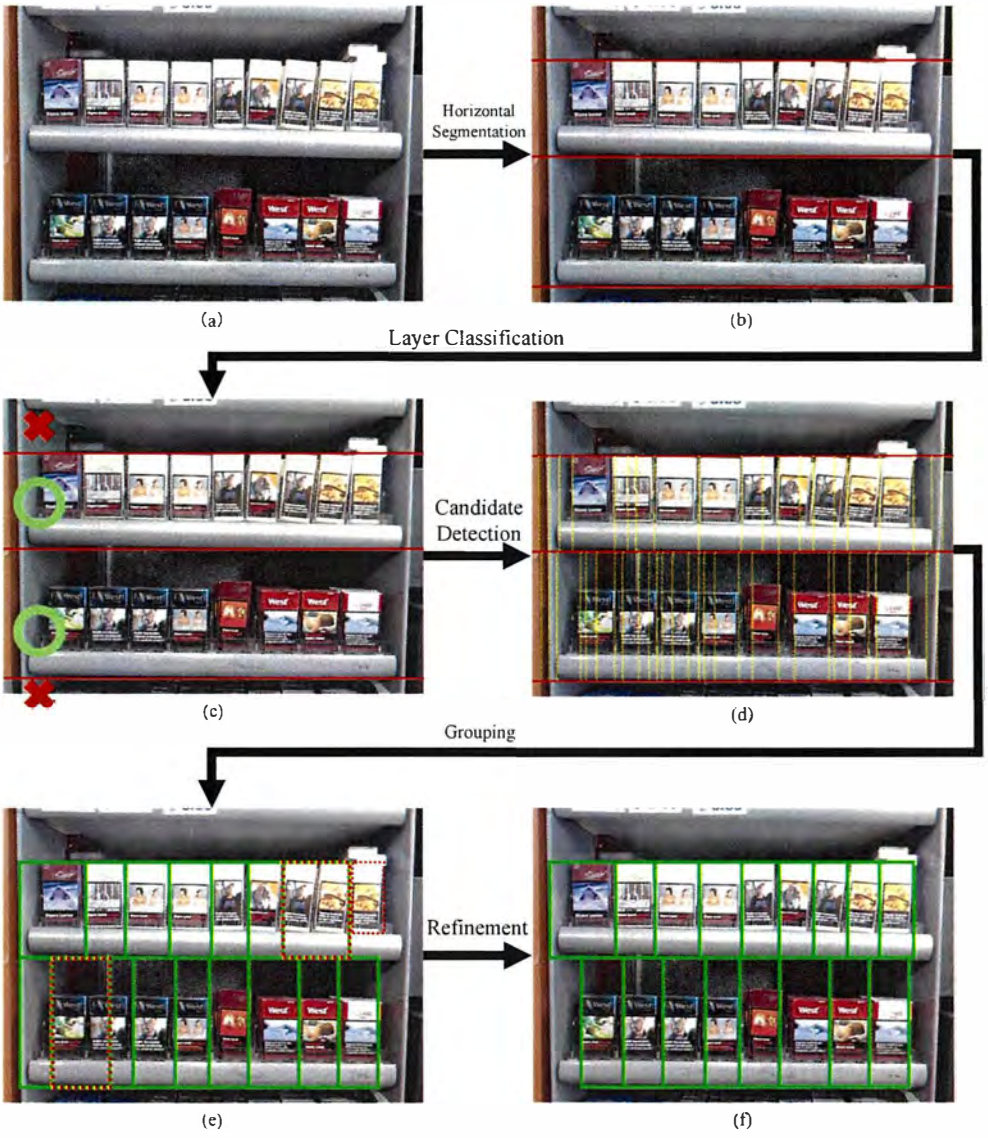


Figure 4.2: The overview of TemplateFree. (a) The retail store shelf image. (b) The detected horizontal separatrices, illustrated by red. (c) The classified layers. The product layers are marked by green circles and the non-product layers are marked by red crosses. (d) The detected vertical separatrix candidates, illustrated by yellow dotted lines. (e) The detected vertical separatrices, with the misdetection and miss-detection illustrated by dotted red bounding boxes. (f) The refined result, where the misdetection and miss-detection in (e) disappear.

Figure 4.3: The flowchart of horizontal segmentation. (a) An example of retail store shelf images. (b) The preprocessed image. (c) The detected horizontal separatrix candidates, illustrated by green on the preprocessed image. (d) The detected horizontal separatrices, illustrated by red. All combinations of the straight lines in (c) are optimized. The best solution to the combinations is treated as the separatrices shown in (d).

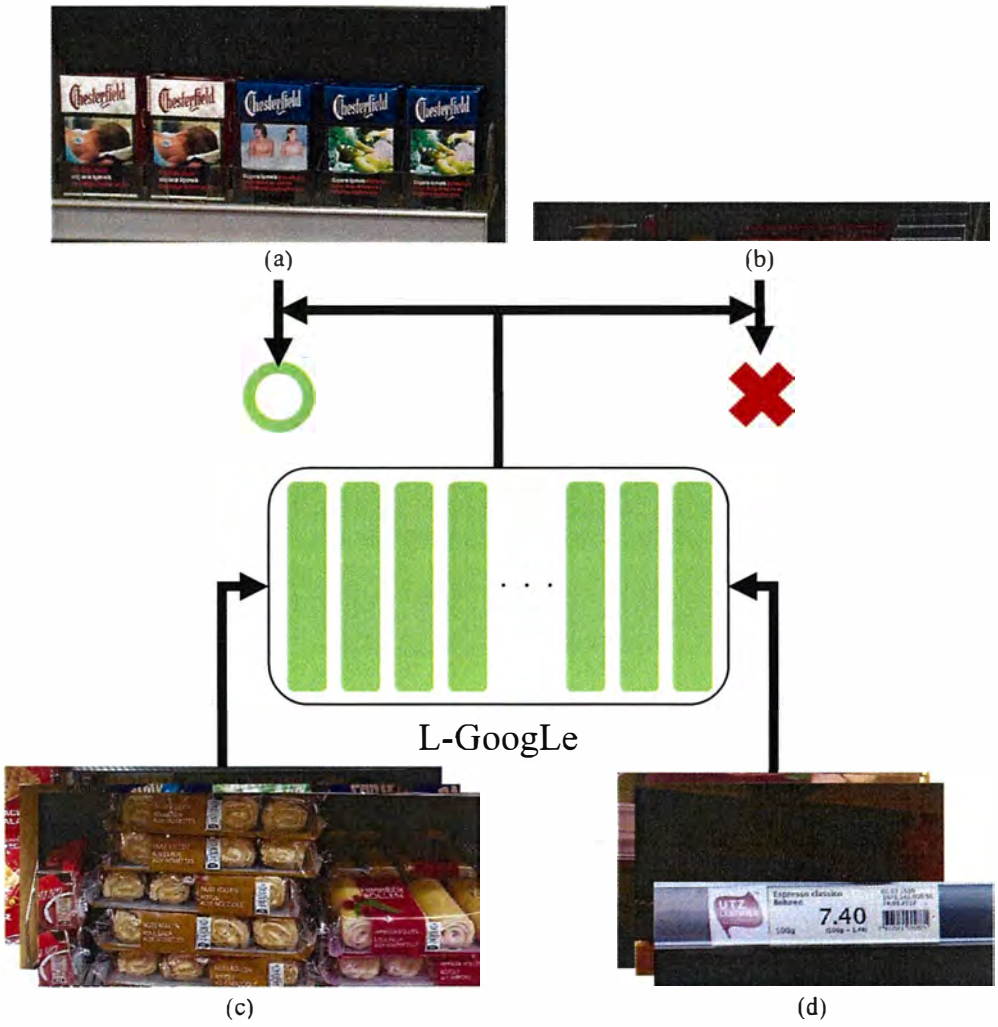


Figure 4.4: The workflow of product region classification. (a) An example of the product layer. (b) An example of the non-product layer. (c) Examples of the positive training data. (d) Examples of the negative training data. The L-GoogLe learns positive and negative data exampled in (c) and (d), then classifies layer images such as (a) and (b). The green circle and red cross show the classification results of (a) and (b) respectively.

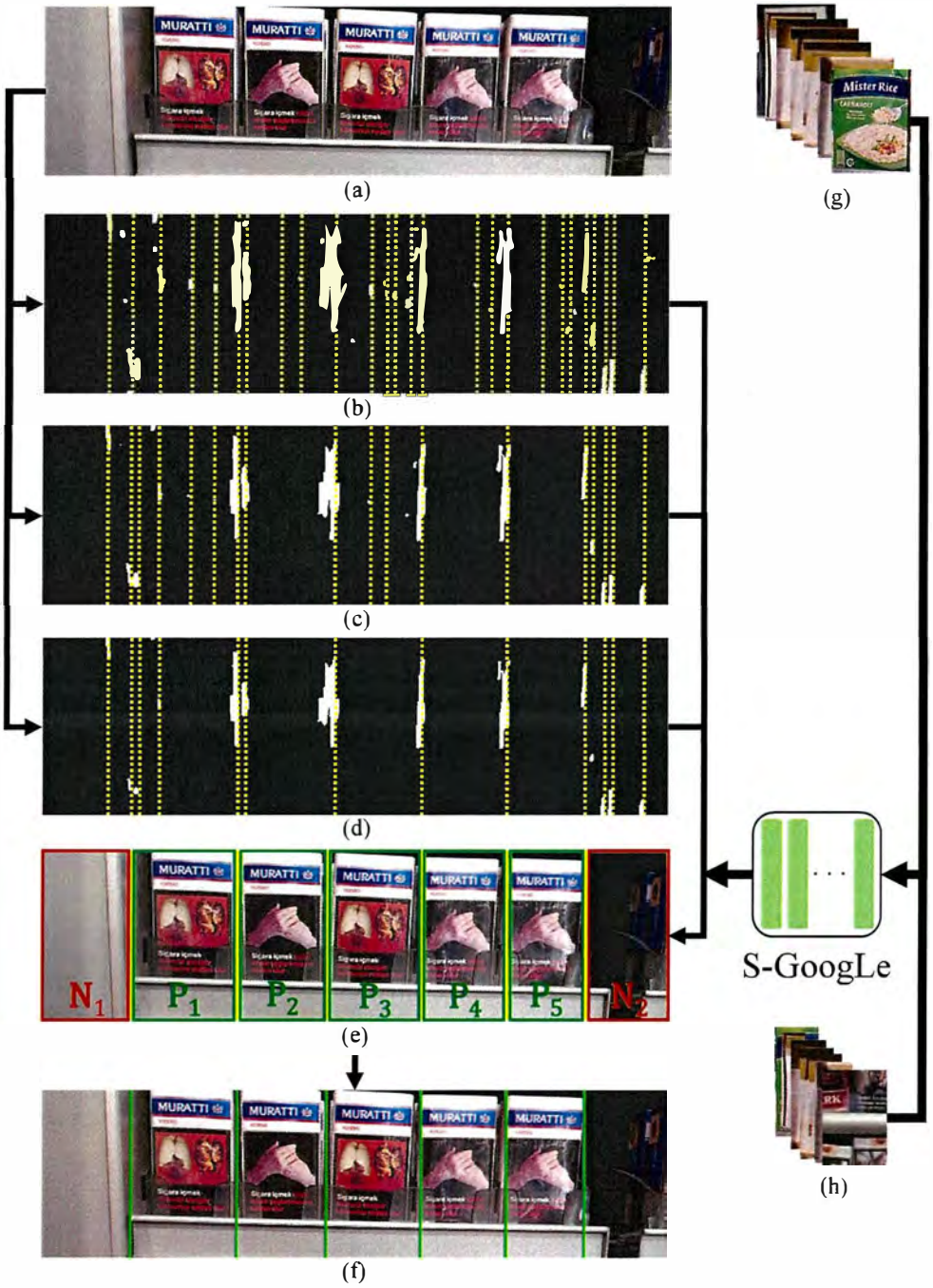


Figure 4.5: The workflow of vertical segmentation. (a) An exampled product layer. (b)~(d) Preprocessed product layer image with different parameters. The candidates are illustrated by dotted yellow lines. (e) An example of the evaluation of grouping. The potential positive regions are within green bounding boxes, denoted by $\mathbf{P}_{1\sim5}$, while the potential negative regions are within red bounding boxes, marked by $\mathbf{N}_1$ and $\mathbf{N}_2$. (f) The detected vertical separatrices, illustrated by green. (g) Examples of the positive training data. (h) Examples of the negative training data.

(a) (b)

Figure 4.6: The comparison between training data and test data. (a) The example of training data. (b) The example of test data. They are completely different in the categories of shelves and products visually.

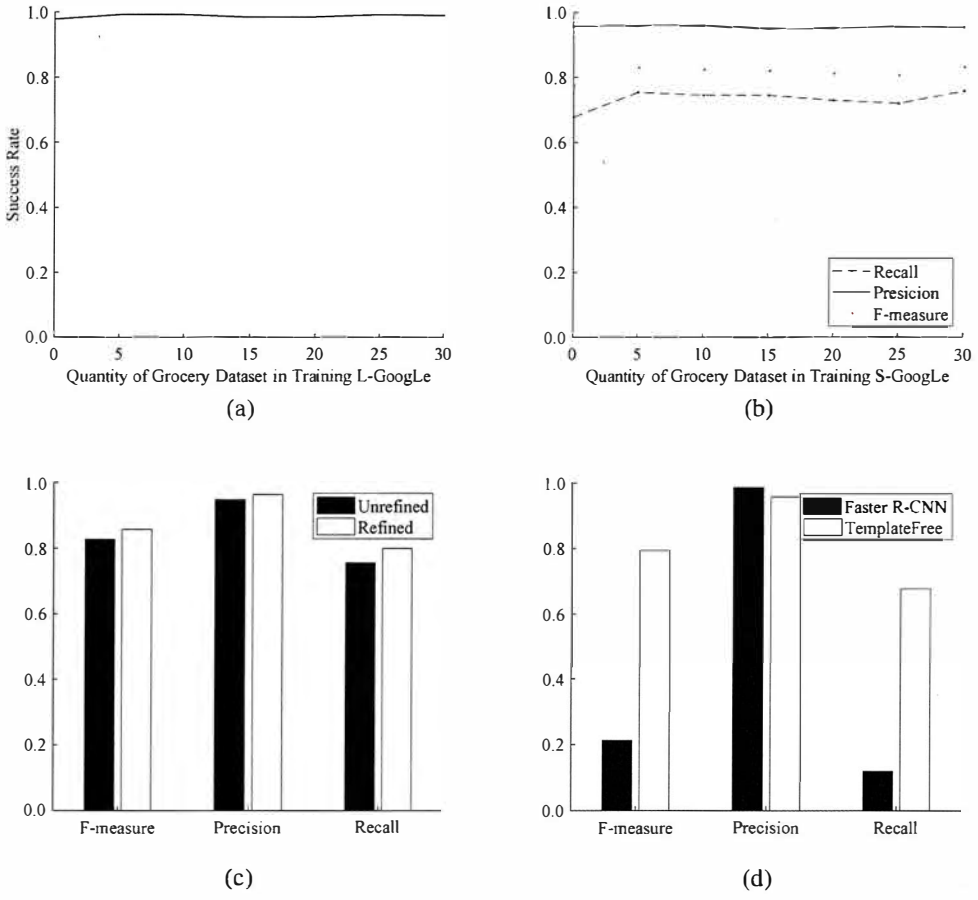


Figure 4.7: The quantitative evaluation. (a) The success rate of L-GoogLe with the quantity of the relevant data increasing. (b) The performance of S-GoogLe with the quantity of relevant training data increases. (c) The evaluation of the refinement. (d) The comparison between TemplateFree and faster R-CNN.

Figure 4.8: Visual comparison between faster R-CNN and TemplateFree. The results of Faster R-CNN are exampled in (a) and (c), and the results of TemplateFree are exampled in (b) and (d). The original images of (a) and (b) are the same, so do (c) and (d). Faster R-CNN performs bad with the training data in different classes, while TemplateFree can detect almost all of the products. In (b), the red crosses mark the false detection and miss detection of TemplateFree.

Chapter 5

Conclusion & Future Work

In this thesis, multiple categories of queries based approaches are proposed, including a recurrent bidirectional VHRP, a novel sketch compression for SBIR, and TemplateFree for product detection. Experiments reveal that the three approaches performs better than the existing methods, improving image retrieval and object detection.

For the commonalities of the queries, on one hand, so far, the three approaches are not applied to one project. Hence, the commonalities of the three categories of queries are simply discussed here. The three approaches can play one role together, shown in Fig. 5.1. E.g., SBIR can be used in "paint to search", becoming the basic of VHPR and TemplateFree. VHPR and TemplateFree can appear together in a commercial analysis system. On the other hand,

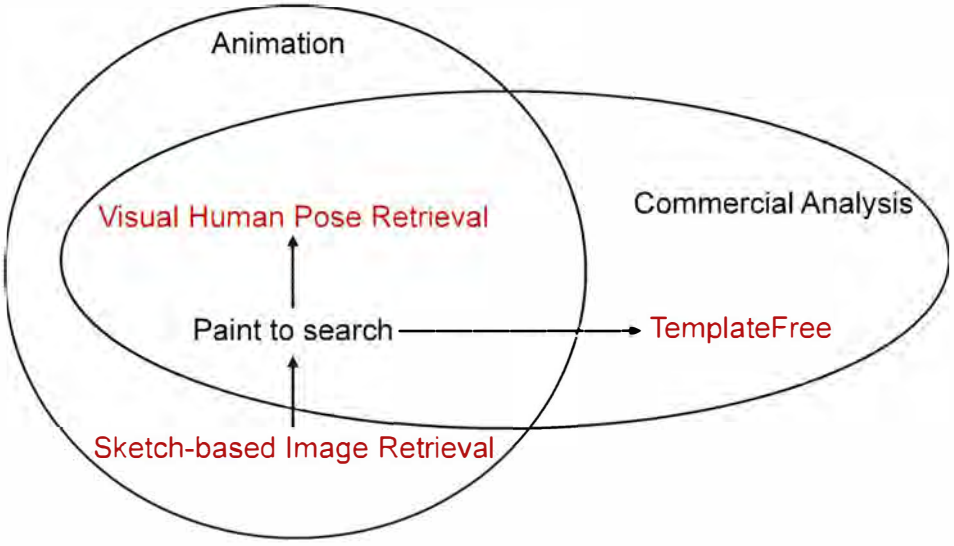


Figure 5.1: Examples of future works.

Bibliography

[1] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision*. Springer, 2006, pp. 404–417.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *The IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 886–893.

[3] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *The ACM International Conference on Multimedia*. ACM, 2014, pp. 157–166.

[4] H. Yu, M. Li, H.-J. Zhang, and J. Feng, "Color texture moments for content-based image retrieval," in *Proceedings. International Conference on Image Processing*, vol. 3. IEEE, 2002, pp. 929–932.

[5] Y. Rubner, L. J. Guibas, and C. Tomasi, "The earth movers distance, multi-dimensional scaling, and color-based image retrieval," in *Proceedings of the ARPA image understanding workshop*, vol. 661, 1997, p. 668.

[6] A. K. Jain and A. Vailaya, "Image retrieval using color and shape," *Pattern recognition*, vol. 29, no. 8, pp. 1233–1244, 1996.

[7] B. S. Manjunath and W.-Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[8] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision*, vol. 1, no. 1-22. Prague, 2004, pp. 1–2.

[9] D. G. Lowe *et al.*, "Object recognition from local scale-invariant features." in *iccv*, vol. 99, no. 2, 1999, pp. 1150–1157.

[10] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov, "Hamming distance metric learning," in *Advances in neural information processing systems*, 2012, pp. 1061–1069.

[11] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *Journal of Machine Learning Research*, vol. 11, no. Mar, pp. 1109–1135, 2010.

[12] H. Chang and D.-Y. Yeung, "Kernel-based distance metric learning for content-based image retrieval," *Image and Vision Computing*, vol. 25, no. 5, pp. 695–703, 2007.

[13] R. Salakhutdinov and G. Hinton, "Semantic hashing," *International Journal of Approximate Reasoning*, vol. 50, no. 7, pp. 969–978, 2009.

[14] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 11–18.

[15] A. A. Ward, M. E. Graham, K. J. Riley, and N. Sheen, "Enhancing a historical digital art collection: Evaluation of content-based image retrieval on collage," *Digital Art History: A Subject in Transition*, vol. 1, pp. 114–125, 2005.

[16] T. Glatard, J. Montagnat, and I. E. Magnin, "Texture based medical image indexing and retrieval: application to cardiac imaging," in *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*. ACM, 2004, pp. 135–142.

[17] B. Ramamurthy and K. Chandran, "Content based image retrieval for medical images using canny edge detection algorithm," *International Journal of Computer Applications*, vol. 17, no. 6, 2011.

[18] P. Conway, "Modes of seeing: Digitized photographic archives and the experienced user," *The American Archivist*, vol. 73, no. 2, pp. 425–462, 2010.

[19] H. Jiang and E. Learned-Miller, "Face detection with the faster r-cnn," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 650–657.

[20] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1904–1912.

[21] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, and W. T. Freeman, "Best-buddies similarity for robust template matching," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2021–2029.

[22] A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran, "Zero-shot object detection," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 384–400.

[23] B. Demirel, R. G. Cinbis, and N. Ikizler-Cinbis, "Zero-shot object detection by hybrid region embedding," in *The British Machine Vision Conference*, 2018.

[24] I. Talmi, R. Mechrez, and L. Zelnik-Manor, "Template matching with deformable diversity similarity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 175–183.

[25] S. Ozekes and A. Y. Camurcu, "Automatic lung nodule detection using template matching," in *International Conference on Advances in Information Systems*. Springer, 2006, pp. 247–253.

[26] R. An, P. Gong, H. Wang, X. Feng, P. Xiao, Q. Chen, Q. Zhang, C. Chen, and P. Yan, "A modified pso algorithm for remote sensing image template matching," *Photogrammetric Engineering & Remote Sensing*, vol. 76, no. 4, pp. 379–389, 2010.

[27] J. Sato and T. Akashi, "Deterministic crowding introducing the distribution of population for template matching," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 13, no. 3, pp. 480–488, 2018.

[28] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[29] ——, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[31] H. Sun, C. Zhang, and T. Akashi, "Recurrent bidirectional visual human pose retrieval," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 14, no. 7, 2019.

[32] G. Qiu, "Indexing chromatic and achromatic patterns for content-based colour image retrieval," *Pattern Recognition*, vol. 35, no. 8, pp. 1675–1686, 2002.

[33] Q. Wang, G. Kurillo, F. Ofli, and R. Bajcsy, "Evaluation of pose tracking accuracy in the first and second generations of microsoft kinect," in *2015 international conference on healthcare informatics*. IEEE, 2015, pp. 380–389.

[34] Y. Jiang, C. Li, and A. H. Paterson, "High throughput phenotyping of cotton plant height using depth images under field conditions," *Computers and Electronics in Agriculture*, vol. 130, pp. 57–68, 2016.

[35] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation." in *bmvc*, vol. 2, no. 4, 2010, p. 5.

[36] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.

[37] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Pose search: retrieving people using their pose," in *The IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1–8.

[38] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari, "2d articulated human pose estimation and retrieval in (almost) unconstrained still images," *International Journal of Computer Vision*, vol. 99, no. 2, pp. 190–214, 2012.

[39] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," *ACM Transactions on Graphics*, vol. 36, no. 4, p. 44, 2017.

[40] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[41] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on Computers*, vol. 100, no. 1, pp. 67–92, 1973.

[42] L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, and B. Schiele, "Articulated people detection and pose estimation: Reshaping the future," in *The IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3178–3185.

[43] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *The IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660.

[44] V. Belagiannis and A. Zisserman, "Recurrent human pose estimation," in *12th IEEE International Conference on Automatic Face & Gesture Recognition*. IEEE, 2017, pp. 468–475.

[45] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 648–656.

[46] W. Yang, W. Ouyang, H. Li, and X. Wang, "End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3073–3082.

[47] H.-J. Lee, C. Zen *et al.*, "Determination of 3d human-body postures from a single view," *Computer Vision Graphics and Image Processing*, vol. 30, no. 2, pp. 148–168, 1985.

[48] C. BenAbdelkader and Y. Yacoob, "Statistical estimation of human anthropometry from a single uncalibrated image," *Computational Forensics*, 2008.

[49] P. Guan, A. Weiss, A. O. Balan, and M. J. Black, "Estimating human shape and pose from a single image," in *The IEEE International Conference on Computer Vision*. IEEE, 2009, pp. 1381–1388.

[50] J. Chen, S. Nie, and Q. Ji, "Data-free prior model for upper body pose estimation and tracking," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4627–4639, 2013.

[51] J. M. Rehg, D. D. Morris, and T. Kanade, "Ambiguities in visual tracking of articulated objects using two-and three-dimensional models," *The International Journal of Robotics Research*, vol. 22, no. 6, pp. 393–418, 2003.

[52] C. Sminchisescu and B. Triggs, "Estimating articulated human motion with covariance scaled sampling," *The International Journal of Robotics Research*, vol. 22, no. 6, pp. 371–391, 2003.

[53] V. Ramakrishna, T. Kanade, and Y. Sheikh, "Reconstructing 3d human pose from 2d image landmarks," in *European Conference on Computer Vision*. Springer, 2012, pp. 573–586.

[54] X. Fan, K. Zheng, Y. Zhou, and S. Wang, "Pose locality constrained representation for 3d human pose reconstruction," in *European Conference on Computer Vision*. Springer, 2014, pp. 174–188.

[55] I. Akhter and M. J. Black, "Pose-conditioned joint angle limits for 3d human pose reconstruction," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1446–1455.

[56] C.-H. Chen and D. Ramanan, "3d human pose estimation= 2d pose estimation+ matching," in *The IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, no. 5, 2017, p. 6.

[57] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[58] R. Ren and J. Collomosse, "Visual sentences for pose retrieval over low-resolution cross-media dance collections," *IEEE Transactions on Multimedia*, vol. 14, no. 6, pp. 1652–1661, 2012.

[59] H. Sun, C. Zhang, and T. Akashi, "Sketch compression for sketch-based image retrieval by linear approximate representation," in *The 23rd Symposium on Sensing via Image Information*, 2017, pp. IS3–11.

[60] ——, "Sketch-based image retrieval: Weighted hog by grayscale value," in *The papers of Technical Meeting on "Perception Information", IEE Japan*, vol. 2015, no. 71, 2015, pp. 17–20.

[61] T. Bui and J. Collomosse, "Scalable sketch-based image retrieval using color gradient features," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 1–8.

[62] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-net: A deep neural network that beats humans," *International Journal of Computer Vision*, vol. 122, no. 3, pp. 411–425, 2017.

[63] J. Song, Y.-Z. Song, T. Xiang, T. Hospedales, and X. Ruan, "Deep multi-task attribute-driven ranking for fine-grained sketch-based image retrieval," in *British Machine Vision Conference*, vol. 3, 2016.

[64] R. Aarthi, K. Anjana, and J. Amudha, "Sketch based image retrieval using information content of orientation," *Indian Journal of Science and Technology*, vol. 9, no. 1, 2016.

[65] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "Sketch-based image retrieval: Benchmark and bag-of-features descriptors," *IEEE transactions on visualization and computer graphics*, vol. 17, no. 11, pp. 1624–1636, 2011.

[66] S. Parui and A. Mittal, "Similarity-invariant sketch-based image retrieval in large databases," in *European Conference on Computer Vision.* Springer, 2014, pp. 398–414.

[67] G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5306–5314.

[68] I. Galić, Č. Livada, and B. Zovko-Cihlar, "Image compression with b-tree coding algorithm enhanced by data modelling with burrows-wheeler transformation," *AUTOMATIKA: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, vol. 57, no. 1, pp. 76–88, 2016.

[69] R. Hu and J. Collomosse, "A performance evaluation of gradient field hog descriptor for sketch based image retrieval," *Computer Vision and Image Understanding*, vol. 117, no. 7, pp. 790–806, 2013.

[70] H. Sun, J. Zhang, and T. Akashi, "Templatefree: Product detection on retail store shelves," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 15, no. 2, 2019.

[71] S. R. Parandker and D. Lokku, "Customer experience management," in *2012 Third International Conference on Services in Emerging Markets.* IEEE, 2012, pp. 44–49.

[72] M.-H. Yang and W.-C. Chen, "A study on shelf space allocation and management," *International journal of production economics*, vol. 60, pp. 309–317, 1999.

[73] S. S. Tsai, D. Chen, V. Chandrasekhar, G. Takacs, N.-M. Cheung, R. Vedantham, R. Grzeszczuk, and B. Girod, "Mobile product recognition," in *Proceedings of the 18th ACM international conference on Multimedia.* ACM, 2010, pp. 1587–1590.

[74] W. Geng, F. Han, J. Lin, L. Zhu, J. Bai, S. Wang, L. He, Q. Xiao, and Z. Lai, "Fine-grained grocery product recognition by one-shot learning," in *2018 ACM Multimedia Conference on Multimedia Conference.* ACM, 2018, pp. 1706–1714.

[75] T. Winlock, E. Christiansen, and S. Belongie, "Toward real-time grocery detection for the visually impaired," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops.* IEEE, 2010, pp. 49–56.

[76] M. Marder, S. Harary, A. Ribak, Y. Tzur, S. Alpert, and A. Tzadok, "Using image analytics to monitor retail store shelves," *IBM Journal of Research and Development*, vol. 59, no. 2/3, pp. 3–1, 2015.

[77] M. George, D. Mircic, G. Soros, C. Floerkemeier, and F. Mattern, "Fine-grained product class recognition for assisted shopping," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 154–162.

[78] A. Tonioni and L. Di Stefano, "Product recognition in store shelves as a sub-graph isomorphism problem," in *International Conference on Image Analysis and Processing*. Springer, 2017, pp. 682–693.

[79] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *Advances in neural information processing systems*, 2009, pp. 1410–1418.

[80] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016, p. 541.

[81] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[82] J. Swartz, S. A. Harrison, E. Barkan, F. Delfine, and G. Brown, "Portable laser scanning arrangement for and method of evaluating and validating bar code symbols," 1981, uS Patent 4,251,798.

[83] D. López-de Ipiña, T. Lorido, and U. López, "Indoor navigation and product recognition for blind people assisted shopping," in *International Workshop on Ambient Assisted Living*. Springer, 2011, pp. 33–40.

[84] M. Kassim, C. K. H. C. K. Yahaya, M. H. M. Zaharuddin, and Z. A. Bakar, "A prototype of halal product recognition system," in *International Conference on Computer & Information Science*, vol. 2. IEEE, 2012, pp. 990–994.

[85] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[86] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[87] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[88] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[89] M. George and C. Floerkemeier, "Recognizing products: A per-exemplar multi-label image classification approach," in *European Conference on Computer Vision*. Springer, 2014, pp. 440–455.

[90] G. Varol and R. S. Kuzu, "Toward retail product recognition on grocery shelves," in *Sixth International Conference on Graphic and Image Processing (ICGIP 2014)*, vol. 9443. International Society for Optics and Photonics, 2015, p. 944309.

[91] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. I–I.

[92] V. Vapnik, S. E. Golowich, and A. J. Smola, "Support vector method for function approximation, regression estimation and signal processing," in *Advances in neural information processing systems*, 1997, pp. 281–287.

[93] E. Gavves, T. Mensink, T. Tommasi, C. G. Snoek, and T. Tuytelaars, "Active transfer learning with zero-shot priors: Reusing past datasets for future tasks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2731–2739.

[94] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *The IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

[95] Z. Zhang and V. Saligrama, "Zero-shot learning via joint latent similarity embedding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 6034–6042.

[96] H. Guo, H. Lin, S. Zhang, and S. Li, "Image-based seat belt detection," in *Proceedings of 2011 IEEE International Conference on Vehicular Electronics and Safety*. IEEE, 2011, pp. 161–164.

[97] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.