Study on Content-Based Copy Detection for Copyright Protection of Line Drawings

Contents

Chapter 1

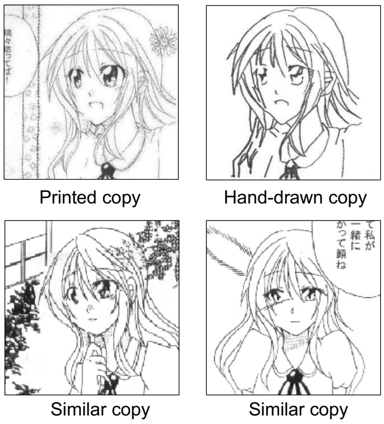
Introduction

Line drawings are a kind of images that consist of distinct straight and curved lines placed against plain background. Typically, they are drawn in black and white or few colors without gradations to represent three-dimension objects. However, because of the simple structure, they are easily understood and able to offer more information directly. Among the image publications, line drawings occupy an important position, such as comics, logos and diagram, which occur in great abundance. Due to the design and drawing contained in line drawings, their contents are valuable and the problem of illegal copies acquires more and more attentions.

In actuality, the determination of what constitutes an illegal copy is a controversial problem and always depends on the judgment of professionals. However, because of the huge volumes of line drawing publications requiring copyright protection, it is impossible to check all images manually. Therefore, we need a method to search copies using computers instead of human. The purpose of this research is to apply computer technique to detect candidate copies and retrieve their original information for assisting professionals ' further judgment.

In the field of publications, comics are the one of the most popular materials, also the most variable ones. In this research, we focus on comic publications, especially manga (Japanese style comics), and study the detection of their copies. The term "manga" is derived from the Japanese word and describes a kind of serial narrative artwork expressed by sequential comics and print cartoons. Normally, a certain manga series consists of multiple volumes containing several hundred manga pages. As shown in Fig. 1.1(a), objects in manga pages are mainly drawn with lines, but tones and word balloons elaborating the story lines are also employed. Although some full-color manga exist, manga publications are typically printed in black-and-white.

(a)



(b)

(c)

Figure 1.1: Examples of copies of manga. (a) shows an example of original manga pages. (b) shows various types of partial copies. (c) shows a similar copy of cartoon character embedded in a complex background.

The modern style of manga developed in Japan in the late 19th century. In its short history, it has developed quickly to become one of the most popular types of image publication in the world. Especially in Japan, manga occupies a pivotal position in the publishing industry. From a report by the AJPEA (All Japan Magazine and Book Publisher 's and Editor 's Association) in 2006, manga accounts for 36.7% of all publications [1]. Moreover, volumes of famous manga have been translated to other languages and exported to many countries. The North American manga market (2007) generated $210 million annual sales [2], and manga market of France and Germany (2006) also reached $212.6 million [3]. With the development of digital techniques, the comics are not only limited in printed publications, but also converted to e-manga (digitalized manga) and distributed through the Internet, which takes up 75% of the entire eBooks' market in Japan [4] . The protability of manga is not only conned to the sales of manga publications; revenues from other areas like advertisements [5] and animation [6] lms using main cartoon characters of manga are also signicant. Therefore, there is great interest in protecting the related copyrights.

For copying manga, illegal users not only duplicate whole manga pages directly, but also focus on certain interesting parts to make partial copies. As shown in Fig. 1.1(b) focusing on a part of the manga page, several kinds of partial copies can be created. Based on the copying method, we classify the copies as follows. Printed partial copies are produced by scanning printed manga and always include some image noise caused by printing. Hand-drawn partial copies are created by tracing the main lines of manga pages, which contain many changes in detail. These two kinds of copies are based on contents of a specific original and thus are called exact copies. In contrast to exact copies, similar copies infringe copyrights of cartoon characters. By imitating their features, new drawings of the same characters are created with different expressions, such as facial expressions, poses or viewpoints. Furthermore, these kinds of partial copies are always applied as parts of other drawings, as shown in Fig. 1.1(c). Therefore, for copyright protection of manga, all kinds of partial copies should be detected from complex backgrounds.

There are few methods proposed for the detection of illegal copies of manga. However, methods applied to other images offer us much related work.

Digital watermarking For protecting the copyright of color images, digital watermarking is a commonly applied method [7]–[10]. By changing the contents of images slightly, copyright owners can incorporate images into their work identifying information. Within huge amounts of color information, watermarks can be embedded invisibly to the human eye. Many watermarking schemes have been designed to resist geometric distortions using such methods as invariant transforms [7], image features [8], image segmentation [9] and template insertion of frequency domain [10]. However, since manga generally have minimal color information, it is difficult to embed enough information without being perceived. In addition, the hand-drawn copies simply trace the main outline of the originals; thus any embedded information will not be included.

Graphic recognition Since manga are drawn in lines, recognition of line drawings, such as engineering drawings [11]–[14], maps [15], flow diagrams [16] and logos [17] is another method we can consider. Depending on shape or line attributes, line drawings are segmented into some primitives and recognized based on a pictorial database. After vectorization, line drawings can be converted to symbols. Based on prior knowledge or some natural characteristics of specific drawings, some systems can achieve better performances [12]–[14]. However, since constructs of manga do not follow a fixed style, it is difficult to divide them into regular components or recognize them depending on some prior knowledge.

Content-Based Copy Detection (CBCD) Since illegal copies contain elements similar to the originals, copy detection using Content-Based Image Retrieval (CBIR) or Content-Based Video Retrieval (CBVR), called Content-Based Copy Detection (CBCD), is another available method for detecting image copies [18]–[22] and video copies [23]–[25]. In this method, copyrighted originals are collected to build a database, suspicious images or videos are retrieved as queries. By matching the features extracted from queries and images in the database, the queries with their similar to the original images are reported to the users. Because there are kinds of variations in the copies, robust image features are important for the retrieval. For kinds of image transformations, since local features are robust, they are popularized in CBCD methods. Mikolajczyk and Schmid [26] have compared various kinds of local features and proved that the SIFT (Scale-Invariant Feature Transform) [27] based descriptors perform best for image retrieval. Ke *et al.* [20] proposed to detect near-duplicates and sub-images based on PCA-SIFT [28] features which are Principal Components Analysis based representations of SIFT features. Douze *et al.* [23] proposed applying SIFT features and a bag-of-features method to detect keyframes of video. Their performance on TRECVID2008 proved the robustness of the method to transformations like cropping, occlusions, etc. However, in the case of manga, hand-drawn copies change lines in detail which issue a challenge to the robustness of these local features. Furthermore, since similar copies are created based on common features of manga characters perceived by human beings, these methods are not available any more.

In this research, we follow the CBCD approach for detecting various copies of line drawings. As shown in Fig. 1.2, the copy detection includes two parts: computer processing and human processing. In the part of computer processing, first, a database of copyrighted manga pages is required. Manga pages for checking, called suspicious images, are treated as queries. Depending on the retrieval, the copies are detected from the queries and reported with the information of inviolate originals as evidences for the further check of human. Finally, whether the copies are illegal or not are judged by professionals. Compared to checking all suspicious images manually, users only need to justify illegal copies in reported ones by using the system.

The work of our research focuses on retrieval, which significantly reduces the the work-

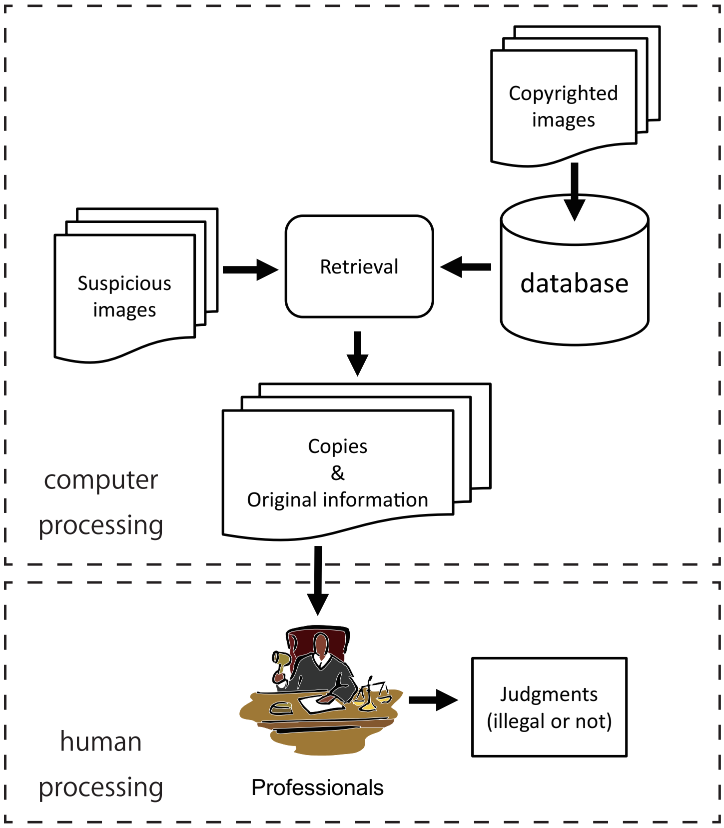


Figure 1.2: Processing of detecting illegal copies.

load of human , while missing few copies. Because of the simple compositions and abstract expressions applied in manga, the unusual types of copies like hand-drawn copies and similar copies issue challenges to the detection. According to the problems, we proposed several methods. In this thesis, we describe these methods and make a summary of them for copy detection of manga. Through the experiments, we prove the effectiveness of the proposed methods and discuss the factors for performance.

The following parts of this thesis are organized as follows.

In chapter 2, the research on exact partial copy detection using local feature matching is described. Through our research, we found that the method using SIFT features is available for detecting the printed copies, but not robust for the detail changes of lines caused by hand drawings. Considering the regions surrounded by lines are relatively robust, we proposed a novel feature based on these regions. In addition, we studied the reliable matchings and proposed a robust matching method. As the results, not only printed but also hand-drawn partial copies can be detected from unknown backgrounds. Meanwhile, their corresponding original parts can also be reported. Moreover, for practical applications, we also studied the reduction of database and feature vector indexing methods, and made an improvement of database construction, by which we achieved a better performance with a smaller database. In the experiments, a database of $10,009$ manga pages were applied for discussion of the proposed method.

In chapter 3, we focus on the problem of similar copies and present our studies on cartoon character detection and recognition based on Regions of Interest (ROIs). Since the similar copies of cartoon characters are created based on the different images of a specific character not a specific part of images, the method described in chapter 2 is not available for their detection. Considering the faces are a basic part of cartoon characters, we propose applying face regions as ROIs for their detection and recognition. First, cartoon characters are located in the suspicious manga pages by ROI detection. Then, the detected cartoon characters are recognized by matching with similar ones in the database. However, since the faces of cartoon characters contain few identifiable features with many variations, such as facial expressions, viewpoints and occlusions, it is difficult to recognize them only depending on features extracted from the face regions. For this problem, we proposed Concentric Multi-Region (CMR) model to explore the significant features from the parts around face regions. Histograms of Oriented Gradients (HOG) [29] is utilized for the description of regions, and the AdaBoost algorithm [30] is applied to obtain a new descriptor named Concentric Multi-Region Histograms of Oriented Gradients (CMR-HOG). To test the effectiveness of the proposed method for cartoon character recognition, 17 labeled cartoon characters are applied in the experiment. Compared to other face and object recognition methods only based on face regions, the proposed method shows a better performance. Furthermore, we make a discussion on scalability of the proposed method for detecting similar copies of cartoon characters by experiments.

In chapter 4, we describe the research on series retrieval of manga. In this work, we study the features of serial manga publications, and proposed a bag-of-features (BOF) approach to retrieve their series. As described in chapters 2 and 3, the CBCD methods require databases of originals. However, since the popular manga or animations are serial publications whose amount are enormous and increasing rapidly, it is difficult to collect the whole series to build the database. Since there are some characteristic parts shared within the whole series, we propose to retrieve the series of unreleased publications (will be released in the future) based on the features from released ones. For this purpose, BOF model is applied. The visual words are based on the features proposed in chapters 2 and 3. We also proposed a method to select discriminative features for the visual word dictionary. In the experiments, we applied typical serial publications : serial manga and animations as data, and proved the proposed method is available for series retrieval of manga pages and animation fragments.

Finally, we make a summary of our research and make an outlook of the copy detection of manga in Chapter 5.

Chapter 2

Partial Copy Detection Using Local Feature Matching

2.1 Introduction

Recently, the prevalence of high-resolution scanners and digital cameras offer us more facilities to store printed articles by converting them into digital images. In contrast to the conveniences, it is easier to make illegal copies and distribute them. Especially for manga publication, the copyright problem is seriously threatening the manga industry. Therefore, there is a great requirement for detecting the copies. Based on the contents of original and suspicious images, the content-based copy detection (CBCD) method can achieve the goal while reducing the workload of human, such as the method in [20]. However, in actuality, illegal users usually try to avoid copyright detection by changing the contents of originals, such as cropping or applying image transformations, which make a difficult to these methods.

In this Chapter, we focus on exact copies which are created based on the specific contents of a certain image. As shown in Fig. 2.1, we classify exact copies into ve types based on the variations contained in the copies. Duplicates, which are intact copies of the whole image, are the easiest to be detected. By applying image modication to the whole image, copies are not exactly the same as the original ones, which are called near-duplicates. In addition, since some illegal users may just copy some interesting parts from the original images, partial copies are created. Partial copies can also be divided into intact and near partial copies by whether applying modications to the original images. From the viewpoint of detection, the case of near partial copies with backgrounds is the most difcult problem.

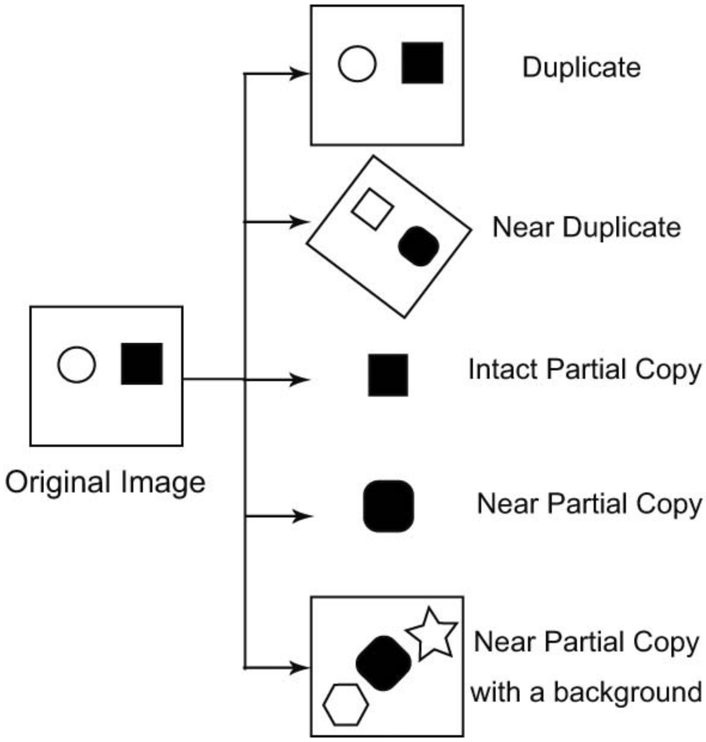
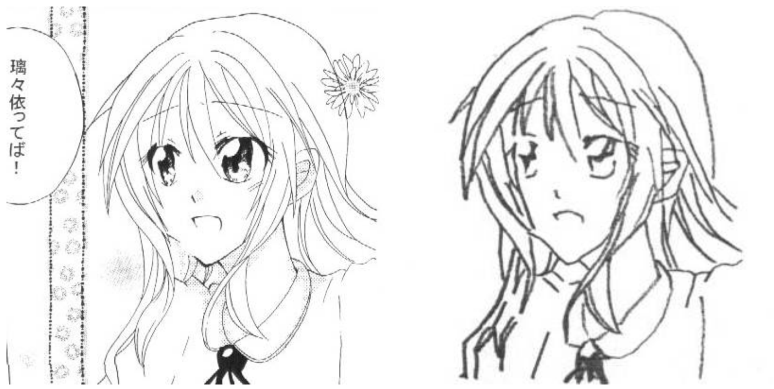


Figure 2.1: Types of exact copies.



(a) (b)

Figure 2.2: Example of hand-drawn copy. (a) shows an original part. (b) shows a hand-drawn copy.

Especially, since line drawings basically consist of lines, hand-drawn copies can also be created by tracing their outlines. As shown in Fig. 2.2, compared to the original part, in the hand-drawn copy, the text balloon and some decorations are removed as well as many changes of lines in detail. All these changes issue a great challenge to the detection.

Different types of copies may require different detection methods. We set the most difficult problem as the target, and propose using local feature matching to detect near partial copies from unknown backgrounds [31]–[34]. Certainly, the proposed method is also available for detecting the duplicates and near-duplicates.

The rest parts of this chapter are arranged as follows: Section 2.2 is the related work. Section 2.3 introduces the proposed method. Experimental results and the discussions are shown in section 2.4. Finally, section 2.5 is the conclusion of this chapter.

2.2 Related work

Since copied contents in illegal copies are similar to their originals, content-based copy detection (CBCD) is an effective method for detecting image copies [18]–[22] and video copies [23]–[25],[35]. In this method, copyrighted images are collected to build a database, suspicious images are treated as queries. By matching the similar parts in the database, copies can be retrieved. Rather than using metadata such as keywords or tags, CBCD methods do the retrievals by analyzing visual content of images, such as color [36],[37], texture [38],[39], and shape [40]–[42]. Thus, it is essential for CBIR systems to have appropriate feature representations of images with corresponding similarity measures to rank the retrieved results.

At the point of feature regions, global attributes [37],[35],[43] have the limitations in resistances to cropping, shifting or compositing, for which features based on distinctive local region detectors show good performances. Local features are therefore often applied in CBCD methods. Berrani *et al.* [18] proposed an image near-duplicate detection system employing local differential descriptors and approximate similarity search. Zhang and Chang [19] used color and texture features around interest points detected by SUSAN corner detector [44] for near-duplicate detection. Ke *et al.* [20] employed PCA-SIFT [28] features for detecting near-duplicates and sub-images. Joly *et al.* [24] proposed a video copy identification method based on Harris interest points [45] and a differential description of the local region around each interest point from key frames corresponding to extrema of the global intensity of motion [46]. Douze *et al.* [23] applied a bag-of-features method to detect key frames of videos, in which SIFT [27] and CS-LBP [47] features are applied to describe Hessian-Affine regions [48].

Because of various kinds of transformations applied and local similarity in copies, to improve the performance of retrieval, reliable matching strategies are carefully designed in many methods. Ke *et al.* [20] employed a typical criterion of similarities and RANSAC

(RANdom SAmple Consensus) [49] for affine geometric verification. Zhang and Chang [19] applied Stochastic Attributed Relational Graphs of interest points. Hsiao *et al.* [21] applied virtual prior attacks to copyrighted images to generate novel features. Zhao and Ngo [22] proposed Scale-Rotation invariant Pattern Entropy to measure the spatial regularity of matching patterns formed by local keypoints.

Since a tremendous amount of features are stored in the database, fast similarity searching is another critical step for CBCD systems. To handle the scalability issue, approximate nearest neighbor search techniques such as [50]–[53], exploring tradeoffs between accuracy and speed have been actively studied. Muja and Lowe [54] proposed applying multiple randomized k-d trees for matching SIFT features. Joly *et al.* [25] proposed an approximate search method with probabilistic selection of feature space regions based on the distribution of the features distortion. In [20] Local Sensitive Hashing (LSH) [51] was employed for indexing features. Rather than using randomized projections in LSH, Weiss *et al.* [53] proposed code generation using the projections on the principal component directions, and Torralba *et al.* [55] applied Restricted Boltzmann Machines [56] and a Boosting method [57] to learn the codes. Both of them were applied to image global features for a database of millions of images.

Our method follows the way of CBCD methods based on local features. The most similar research is [20], which offered a good performance on sub-image and near copy detection of color images. However, in the case of manga, hand-drawn copies pose a challenge to this method. Our previous work [32] introduced a feature matching method to detect partial copies of line drawings. It achieved detection of printed and hand-drawn partial copies of manga, and has been proved to be robust to rotations and scale transformations under a certain range. In [33], we proposed indexing the feature vectors by a hash table to speed up the detection. Furthermore, we propose a process to build a more discriminative database with a smaller size and compared several approximate nearest neighbor searching methods in [58].

2.3 Proposed Method

2.3.1 Overview

As shown in Fig. 2.3, the process of the proposed method is divide into two parts. In the database processing, first, the images while require copyright protection are collected.

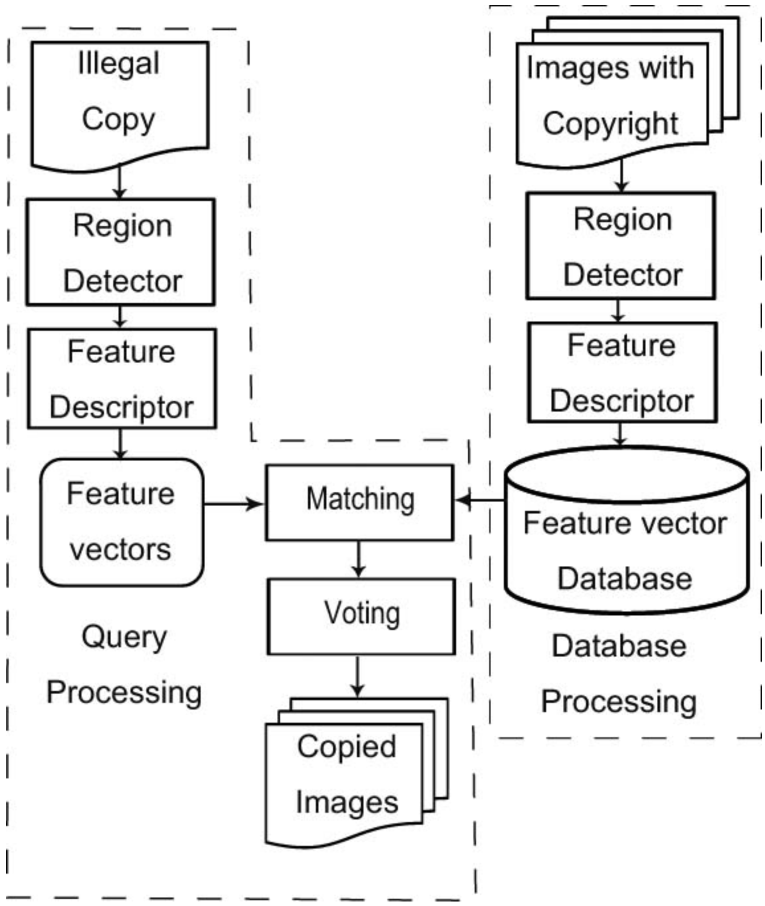


Figure 2.3: Processing of detecting exact partial copies.

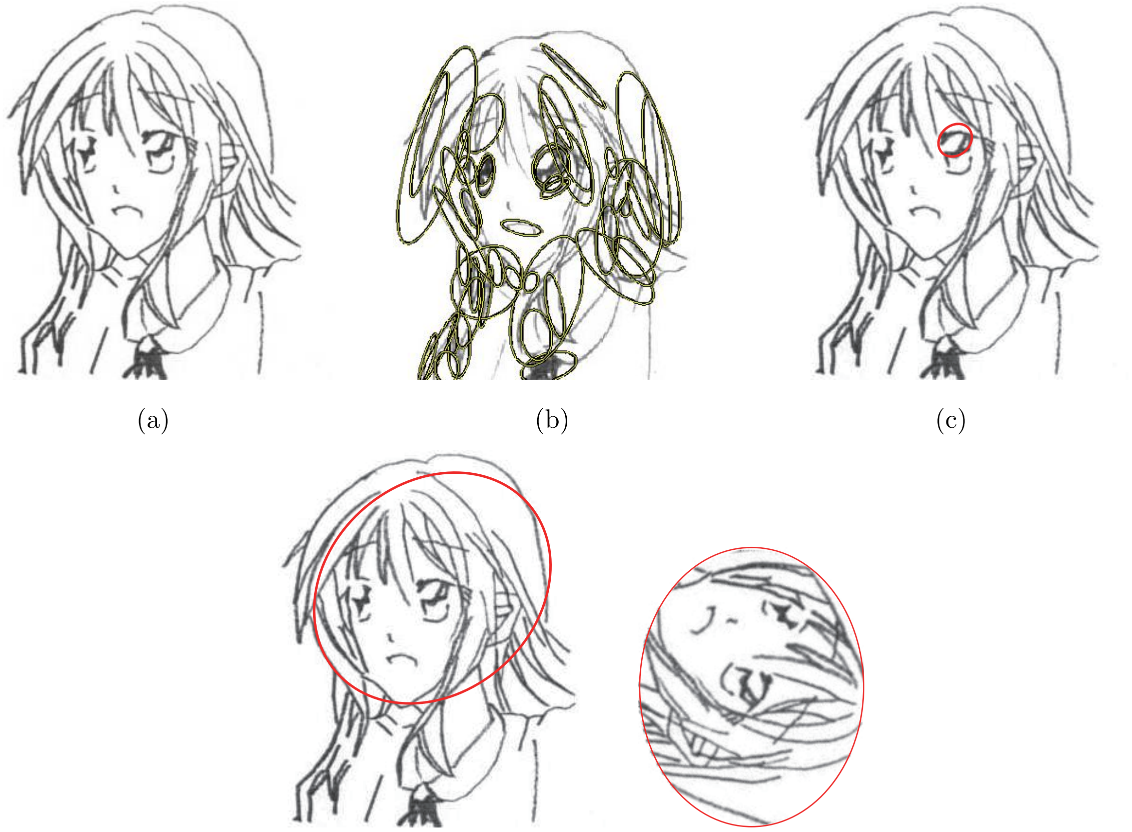
Then, local feature regions (LFRs) are extracted from images by a region detector. After that, through a feature descriptor, LFRs are represented by vectors, called local feature vectors. Based on the copyrighted images, we build a database of local feature vectors with their image labels. In the query processing, a suspicious copy is treated as a query. By the same region detector and feature descriptor, local feature vectors are extracted from the query. In the part of matching, local feature vectors of the query are matched with their nearest neighbors in the database. Through the voting database images by matched feature vectors, the query image with enough votes will be reported as copies with the corresponding database image as evidence.

2.3.2 Region Detector

The region detector is to detect local feature regions (LFRs) from images with the primary goal of obtaining robust, stable and well-defined image features for the further recognition and retrieval. Usually, LFRs are based on interest points, which are detected as corners [44],[59] or blobs [27],[60] of the objects. However, as shown in Fig. 2.2, lines of the hand-drawn copy contain many local variations in thickness, intensity and orientation, it is difficult to obtain robust interest points from lines. Considering the regions surrounded by lines are relatively robust even for the changes of lines, we propose to apply these regions as LFRs. The intensities of these regions are lower or higher than the lines around, thus, we propose applying the algorithm of MSER (Maximally Stable Extremal Region) [61] to detect them.

In MSER algorithm, first, pixels of the detecting image are sorted by intensity. Like a binarization, pixels under a threshold are marked. With growing of the threshold, more pixels are marked and merged with neighbors as regions using the union-nd algorithm [62]. During this process, the function of relative changes between the regions and thresholds is stored. MSERs (Maximally Stable Extremal Regions) are detected as regions corresponding to thresholds which generate local minimums of the function. The MSERs can also be understood as the parts of the image where local binarization is stable over a certain change of intensities. Detection of MSER is related to thresholding, since every extremal region is a connected component of a thresholded image. However, no global or optimal threshold is sought, all thresholds are tested and the stability of the connected components evaluated.

To make LFRs more robust for changes, we approximate the MSERs by diagonalizing their covariance matrices. As shown in Fig. 2.4(b), by this processing, we can obtain some ellipse MSERs from the image. Since the hand drawing may change the image in detail, we filter small regions out as unreliable ones. In addition, as shown in Fig. 2.4(d), to increase the information amount in LFRs, we scale the ellipses of MSERs S_g times. For the setting of S_g, we discuss its relation with performance of copy detection in section 2.4.2. Finally, as shown in Fig. 2.4(e), by rotating the long axis of the ellipse parallel to the y axis of the image, we obtain one LFR.



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

Figure 2.4: Processing of LFR detection. (a) shows one part of manga page. (b) shows MSERs detected from manga. (c) shows an example of reliable MSERs. (d) shows a magnified MSER. (e) shows a LFR.

2.3.3 Feature Descriptor

Then, by the feature descriptor, each LFR is represented by a feature vector. To cope with variations contained in copies of manga, we apply HOG (Histograms of Oriented Gradients) [29] to describe the LFRs. HOG features are often employed for detection of objects such as pedestrians. It is robust to changes of objects' edges and illuminations.

The basic idea of HOG is that local object appearance and shape can often be characterized rather well by the distribution of edge directions, even without precise knowledge of the edge positions. The calculation of HOG feature vectors is shown in Fig. 2.5. The parameters are set based on the discussion in [29]. First, we calculate the gradient magnitude (m) and orientation (θ) at each pixel as follows:

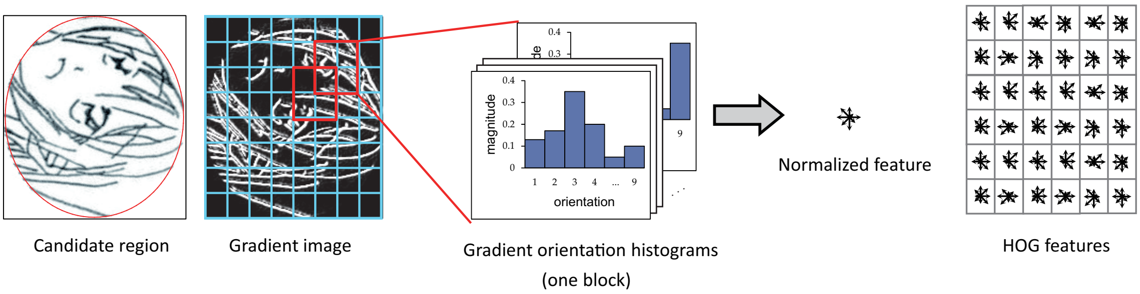


Figure 2.5: Calculation of HOG features.

$$m(x,y) = \sqrt{f_x(x,y)^2 + f_y(x,y)^2} \tag{2.1}$$

$$\theta(x,y) = \tan^{-1}\frac{f_y(x,y)}{f_x(x,y)} \tag{2.2}$$

$$\begin{cases} f_x(x,y) = L(x+1,y) - L(x-1,y) \\ f_y(x,y) = L(x,y+1) - L(x,y-1) \end{cases} \tag{2.3}$$

where $L(x,y)$ is the intensity of pixel (x,y). The gradient orientations are quantized into 9 bins. Then, each LFR is divided into 8×8 cells evenly. For each cell, we calculate the gradient orientation histogram based on the gradient magnitude. Cells are combined into overlapping blocks as 3×3 cells per block, thus 6×6 blocks per ROI. After that, the features (v) is normalized in units of blocks as follows:

$$v = \frac{v}{\sqrt{(\sum_{i=0}^{k} v_k)^2 + 1}} \tag{2.4}$$

where k is the number of features in one block. Thus, we can obtain $9\times3\times3\times6\times6 = 2,916$ HOG features for one LFR. These features are connected into one HOG feature vector of $2,916$ dimension.

To increase the calculation speed of HOG vectors, we apply the integral image to calculate the gradient orientation histograms as in [63]. For each bin we calculate one integral image of gradient magnitude. Every pixel of integral image express the sum value of the left-top rectangle. By 3 times of calculation, we can obtain the sum value of one bin for any cell. Therefore, the orientation histograms containing 9 bins of any cell requires $3 \times 9 = 27$ times of calculation.

Moreover, PCA (Principal Component Analysis) [64] is applied to compress the HOG vectors. We build the eigenspace using $60,000$ training feature vectors and retain the top 100 principal components where the contributing rate is 64%. Finally, the HOG feature vectors are reduced to 100 dimensions, which are called as PCA-HOG feature vectors.

2.3.4 Database Construction

For each LFR, the feature vector with the image ID and the position of its source are stored into databases. Since LFRs are extracted from the whole manga pages, many widely applied patterns, such as tones, words and frames, are extracted as LFRs but without any contribution for copy detection. We propose removing them by the relationship of LFRs in feature vector space instead of using various kinds of filters. For a certain type of patterns, because they follow the similar or almost the same style or shape characteristics, their features vectors are close to others as a cluster. As shown in Fig. 2.6, we detect such clusters and remove them in the database. Since the results of exact copy detection are reported by feature matching between queries and manga pages in the database (described in section 2.3.6) and the similar features of the same page can offer clues for locating the copies, we cluster the features based on features of different manga pages. Specifically, for each feature vector, distances with its neighbors of different manga pages are calculated. If the distance between two feature vectors are less than a threshold T_c, they are similar with each other. By the union-find algorithm, we obtain the clusters of similar feature vectors. Any clusters contain more than one component will be removed. For larger T_c, more feature vectors will get together and consist in one clusters. Therefore, by changing T_c, the number of removed feature vectors can be controlled.

2.3.5 Matching

The similarities of LFRs are rated according to the Euclidean distances of their feature vectors. Each LFR from the query can only be matched with its nearest neighbor in the database, iff the matching satisfies the followings. Denote $\boldsymbol{q}$ as the feature vector extracted from a LFR of a query image, $\boldsymbol{p}_1$ as the nearest neighbor of $\boldsymbol{q}$ in the database, $d(\boldsymbol{q}, \boldsymbol{p}_1)$ is the distance between them, the matching should satisfy $d(\boldsymbol{q}, \boldsymbol{p}_1) < T_g$, where T_g is a threshold. In addition, we apply the matching strategy of distance ratio between the first and second nearest neighbors like [26] to make the matching more discriminative.

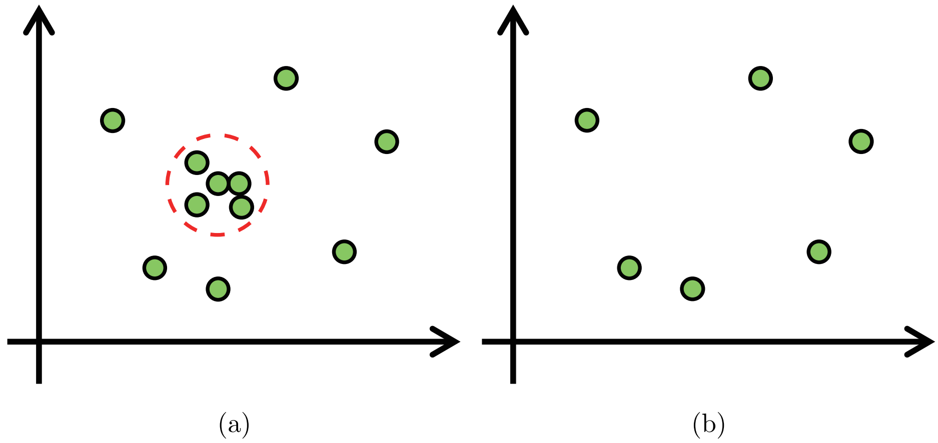


Figure 2.6: Feature spaces of LFRs. (a) shows the feature space of all extracted LFRs. (b) shows the feature space of LFRs applied in the database.

Thus, $\boldsymbol{q}$ and $\boldsymbol{p}_1$ are matched if $d(\boldsymbol{q}, \boldsymbol{p}_1)/d(\boldsymbol{q}, \boldsymbol{p}_2) < R_g$, where $\boldsymbol{p}_2$ is the second nearest neighbor of $\boldsymbol{q}$ and R_g is a threshold.

Furthermore, we employ ANN (Approximate Nearest Neighbor Search) [50] to speed up the matching. ANN is a method to find approximate nearest neighbors by using the k-d tree. First, the feature vector space of the database is subdivided into a k-d tree structure. One leaf of the k-d tree includes only one feature vector and corresponds to the segmented area in the feature space. Figure 2.7 shows an example where rectangles represent leaves of the k-d tree. Given a query vector $\boldsymbol{q}$, by using the k-d tree, the leaf that also contains $\boldsymbol{q}$ is located and the feature vector in this leaf area is regarded as a tentative nearest neighbor with the distance r. The true nearest neighbor is in the hypersphere with this radius. Thus all leaf areas that overlap with the hypersphere are searched and the feature vector with the minimum distance is reported as the nearest neighbor. The approximation of search is done by shrinking the radius r by the factor $1/(1+\varepsilon)$. Although the processing can speed up the matching drastically, shrinking the space is always with the risk of missing the true nearest neighbor. In the case of Fig. 2.7, $\boldsymbol{p}_2$ is found as the approximate nearest neighbor. As shown by the dotted circle, the true nearest neighbor $\boldsymbol{p}_3$ is missed. In the experiments (section 2.4.4), we compare the performance of ANN with other two hashing-based approximate nearest neighbor search methods.

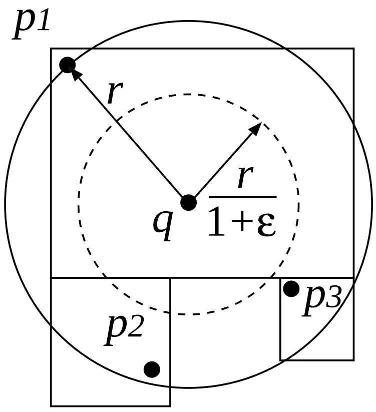


Figure 2.7: Searching process by ANN. Rectangles represent leaves of the k-d tree. q is the query vector. p_1 is the feature vector in the same leaf as q is in, whose distance to q is r. p_2 is the approximate nearest neighbor found by shrinking the radius as $\frac{r}{1+\varepsilon}$. p_3 is the exact nearest neighbor to q, which is missed by the approximate nearest neighbor searching.

2.3.6 System Output

Now, we obtained some LFRs from query images matched with their corresponding LFRs in the database. Based on their manga page IDs and position information, the proposed method can locate the partial copies in the query with copied original parts. We apply affine geometric verification using RANSAC [49] to matched LFRs between queries and corresponding database manga pages. If the number of the verified matchings is over a threshold M_g, the partial copy will be reported.

2.4 Experimental Results

2.4.1 Condition

As data for experiments, we made a collection of 21 titles [1] each of which contains 4 volumes. All the experiments were done using a computer with CPU INTEL Core i7-870 2.93 GHz and RAM 8 GB. The detection results are reported by using recall $R = A/B$ and precision $P = A/C$, where A is the number of correctly detected partial copies, B is the number of correct partial copies, and C is the number of detected partial copies. The recall and precision will change with the threshold T_g for reliable matchings and M_g for outputs. We report the performance of detections by interpolated precision-recall graphs. Since high recall is important to illegal copy detection, for evaluations of performance and analyses of experimental results, we focus on high recall with an acceptable precision.

For exact partial copies, we selected 109 manga pages containing large entire scenes. From each of them, we cropped one part of fixed size (400×400 pixels) that includes some objects interesting to copy. Regarding cropped parts as copy targets, we printed them as printed partial copies and traced their main lines to make hand-drawn partial copies. Then, two kinds of partial copies are scanned into digital images (300×300 pixels). Some examples are shown in Fig. 2.8. For backgrounds, we chose 9 manga pages (about 800×1200 pixels) from our collection as backgrounds, which were not included in the databases. Each exact partial copy was embedded into these backgrounds at random positions, thus $109 \times 9 = 981$ partial copies with backgrounds were created for each type of exact copies.

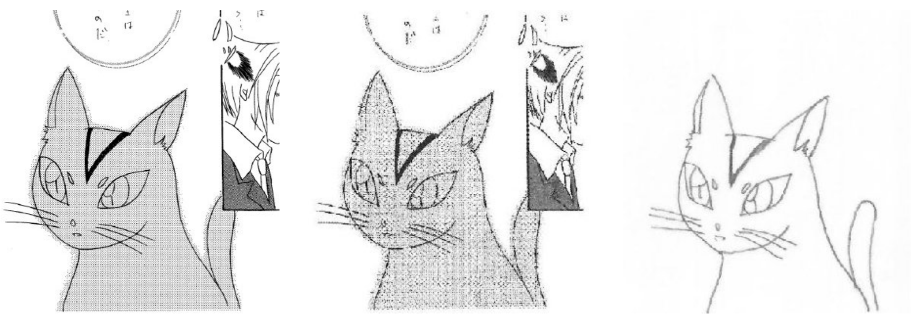
There are four experiments for exact copy detection. The first three experiments are (1) discussion of the scales of LFRs in section **??**, (2) evaluation of feature representations in section **??**, and (3) comparison of searching methods in section **??**. In these three experiments, we applied printed and hand-drawn partial copies without backgrounds as queries and a small database of 109 manga pages, from which partial copies were made. In section **??**, we tested the effectiveness of database construction method by using the partial copies with backgrounds and a database of $10,009$ manga pages. Finally, we make a discussion in section **??**.

[1]The manga applied in experiments include "20th Century Boys", "ARIA", "Kare Kano", "Neon Genesis Evangelion", "Hoshin Engi", "H2", "Hunter × Hunter", "Stone Ocean", "Lucky Star", "Master Keaton", "Maison Ikkoku", "Miyuki", "Monster", "Rozen Maiden", "Planetes", "Rosario + Vampire", , "SLAM DUNK" "Rurouni Kenshin", and 3 titles of manga drawn for this research.

(a) Original part (b) Printed copy. (c) Hand-drawn copy.

(d) Original part (e) Printed copy. (f) Hand-drawn copy.

(g) Original part (h) Printed copy. (i) Hand-drawn copy.



(j) Original part (k) Printed copy. (l) Hand-drawn copy.

Figure 2.8: Example of exact partial copies.

2.4.2 Scale of Local Feature Regions

First, we chose the scale S_g of LFRs depending on precision of matchings. We de-~~tected printed and hand-drawn partial copies with LFRs of different scale as S_g =~~

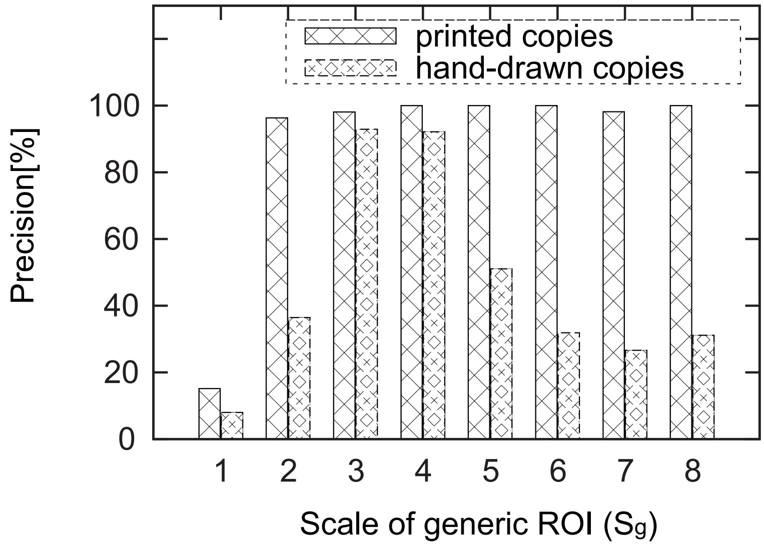


Figure 2.9: Precision at 95% recall for different scales of LFRs.

$1, 2, 3, 4, 5, 6, 7, 8$. For checking the performance of LFRs precisely, exact nearest neighbor searching was applied in this experiment. The threshold of distance rate R_g was set to 0.9 empirically. Based on the threshold of distance T_g, LFRs from partial copies were matched with their exact nearest neighbors, and the precision decreased with increase of recall by changing T_g. We employed the precision at 95% recall to evaluate the performance. M_g was set to 1, thus the results were precision for single ROI.

The results are shown in Fig. 2.9. For printed partial copies, except $S_g = 1$, other scales achieved precisions over 95%. This is because detected MSERs (LFRs of $S_g = 1$) were so small that less discriminative information were contained. For hand-drawn partial copies, $S_g = 3$ performed the best. With increase of the scale, there are more information inside the LFRs including the variations subjected to drawing by hands. We used $S_g = 3$ for the rest of experiments.

2.4.3 Evaluation of PCA-HOG Feature

Next, we tested the effectiveness of the proposed method (named PCA-HOG) comparing with the method using PCA-SIFT features [28] (named PCA-SIFT)

PCA-SIFT Many researches applied PCA-SIFT features and achieved to retrieve images with various transformations [20], [65]. PCA-SIFT features were based on SIFT feature points [27] but use a PCA-based representation. We retrained the eigenspace using $55,000$

patches from manga pages. The process of the PCA-SIFT method was the same as the proposed method except for the employed local features. For the PCA-SIFT method, the parameters of distance restriction, the distance rate and number of verified matchings are T_s, M_s and R (fixed to 0.85 for performance), respectively.

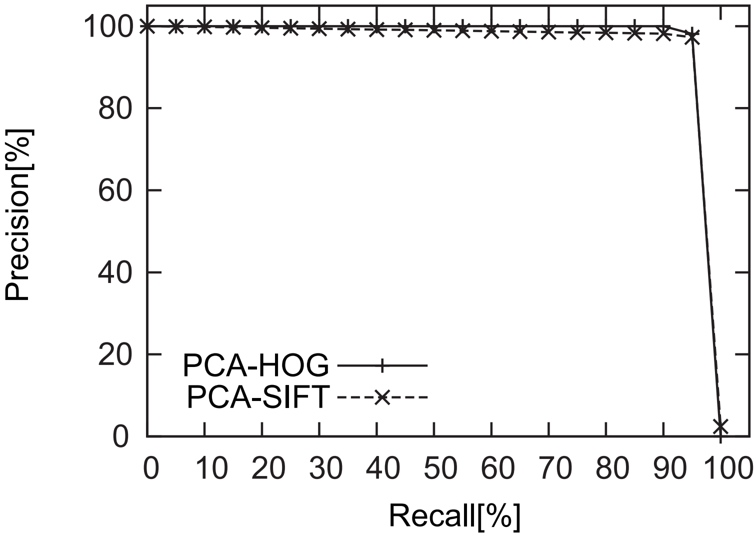
We detected printed and hand-drawn partial copies by exact nearest neighbor search. With changing T_g, M_g for PCA-HOG and T_s, M_s for PCA-SIFT, we obtained interpolated precision-recall graphs, as shown in Fig. 2.10. We can see that both methods had good performance for printed partial copies. However, for hand-drawn copies, compared to the ineffectiveness of PCA-SIFT, PCA-HOG achieved the much better detection. From this experiment, we can see that (1) PCA-SIFT features are not robust for the variations of hand drawings, (2) PCA-HOG features are available for the detection of both printed and hand-drawn copies.

2.4.4 Comparison of Approximate Nearest Neighbor Searching Methods

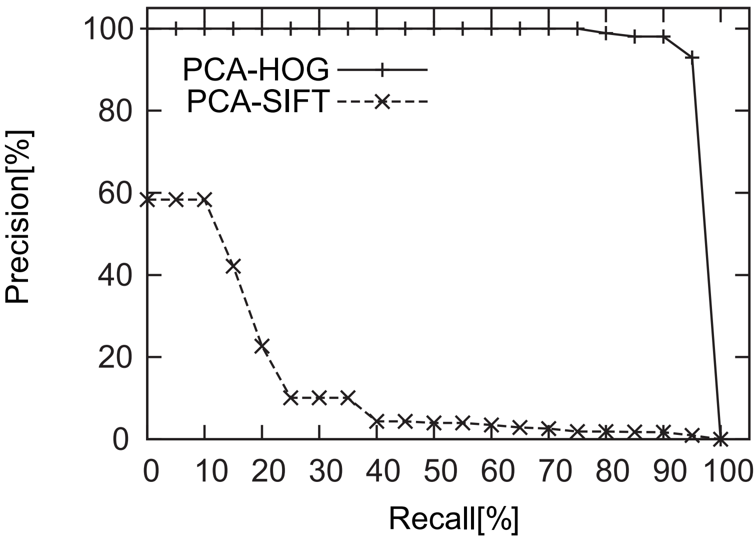
Then, we compared several approximate nearest neighbor search methods for exact partial copy detection, including ANN [50], a method using Spectral Hashing (named SH) [53], and a modified method of SH (named PSH), which only searches in a reliable range using codes of SH.

SH The basic idea of spectral hashing is to map feature vectors to binary codes. Comparing to LSH [51] using randomized projections, spectral hashing generates much more compact codes (length of the code l) by thresholding some nonlinear functions on the projections along the principal component directions. It has been proved to outperform LSH for retrieve similar neighbors [53]. We speed up the searching by retrieve the nearest neighbor within a small Hamming distance d of the code for the query.

PSH Considering the changes of binary codes caused by variations from queries, we also applied a similar strategy as [65] to the SH method. To obtain the binary code, the projection value of each bit was thresholded in the method of spectral hashing. In general some values are very close to the thresholds and likely to be unreliable. Therefore, we evaluated the reliability of each bit by the difference between the projection value and the threshold, and sorted the bits of binary codes in descending order of reliability. The top b bits within a tolerance e are treated as unreliable ones, otherwise reliable. The search was done with the codes whose reliable bits are the same to the code for the query, and



(a) Printed partial copy.



(b) Hand-drawn partial copy.

Figure 2.10: Experimental results of printed and hand-drawn partial copy detection based on PCA-HOG and PCA-SIFT.

other bits were flipped to generate all possible 2^b bit patterns. Therefore, we can only focus on the unreliable parts of codes and speed up the searching further than searching within Hamming distance ranges.

We tried several parameters for the three methods as shown in Table. **??**. Except for searching methods, other settings of the detection are the same to the last experiment. We used precision at 95% recall as criterion for evaluation of detection accuracy. The detection time with corresponding accuracy are shown in Fig. 2.11. We can see the PSH performed

Table 2.1: Parameters of approximate nearest neighbor searching methods.

method	paramter	values
ANN	ε	1,3,5,7,10
SH	l	16,18,20,22,24,26,28,30,32,34,36
	d	0, 1, 2, 3
PSH	l	16,18,20,22,24,26,28,30,32,34,36
	b	8,9,10,11,12,13
	e	0.1,0.2,0.3

better than SH. Compared to ANN, PSH could achieve the same performance for printed partial copies. However, for handwritten partial copies, PSH did not perform better than ANN. This is because hand drawing causes many changes to the codes. As approximate nearest neighbor search, methods based on hashing can find similar data point quickly but may lose many exact nearest neighbors. As the results of [20], 71% correct matchings are missed by using LSH. For the case of many correct matchings contained in queries, the large fraction of missed correct matchings have few affects on the results. Since our hand-drawn partial copies contain few information with many variations, the correct matchings are important for the performance. Therefore, we applied ANN as approximate search method and set $\varepsilon = 5$ for the balance of accuracy and searching speed in the next experiment.

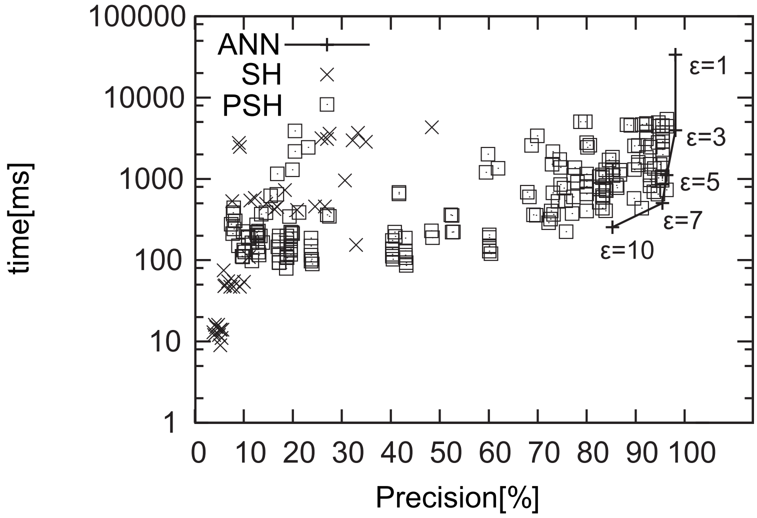
2.4.5 Partial Copy Detection from Complex Backgrounds

In this experiment, we tested the performance of the proposed method for detecting exact partial copies from backgrounds based on larger databases. The queries were 981 manga pages with one partial copy inside each at a random position (named printed copies and hand-drawn copies for short in the following). There were 269 PCA-HOG features per query image on average. The database contained $10,009$ manga pages, from which we obtained $6,423,467$ LFRs and built a database (DB1) using all of them. We empirically set Tc for database construction of the proposed method, and obtained a database (DB2) of $4,338,910$ LFRs.

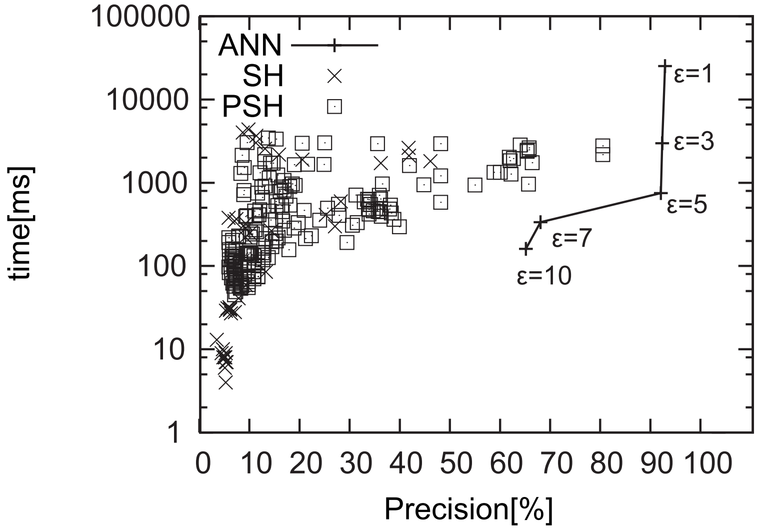
The interpolated precision-recall graphs of the detection based on these two databases are shown in Fig. 2.12. The results show that the performance for both printed and hand-drawn copies were improved by using the process of database construction. The proposed method achieved the detections as 99% precision at 95% recall for printed copies, and 42% precision at 85% recall for hand-drawn copies. Some correctly detected examples are shown in Fig. 2.13 and Fig. 2.14, from which we can see that although there is only a small part of partial copy (about 10% of the whole image) in queries, the proposed method can locate them and offer the copied parts precisely.

For the failures of hand-drawn copy detection, main reasons are listed in the following:

- Few discriminative information. No results were retrieved for some partial copies. Such as the example shown in Fig. 2.15(a), since only the outlines in a small area were traced, few features were extracted. With the variation caused by hand drawing,



(a) Printed partial copy.



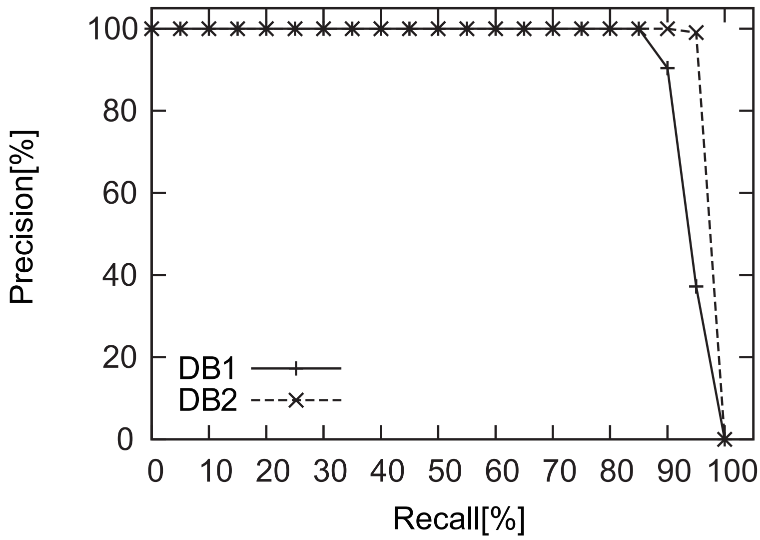
(b) Hand-drawn partial copy.

Figure 2.11: Experimental results of approximate nearest neighbor searching methods (ANN, SH, PSH).

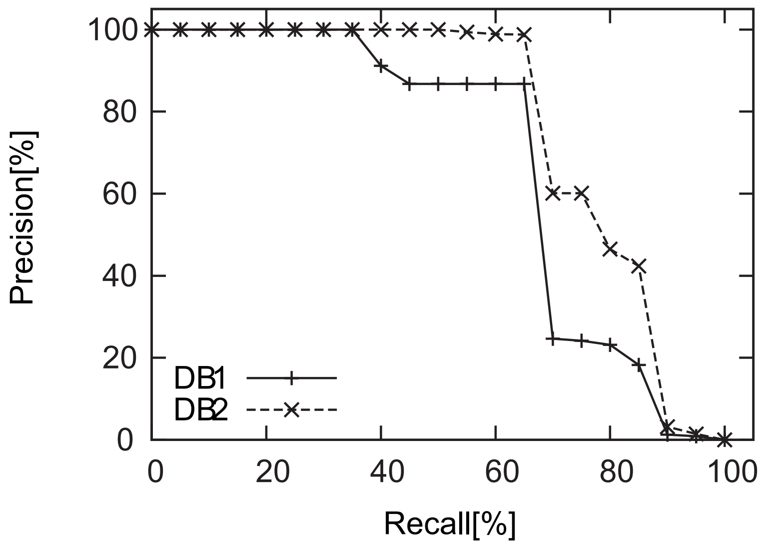
the number of correct matchings was limited.

- Similar parts. Although the manga pages of backgrounds were not included in the database and we also remove the commonly applied patterns by the process of database construction, some parts were still similar and led erroneous matchings, as shown in Fig. 2.15(b) and (c).

The database size (DB2) was 4.1 GB (including the feature vectors, image IDs, and



(a) Printed partial copy.



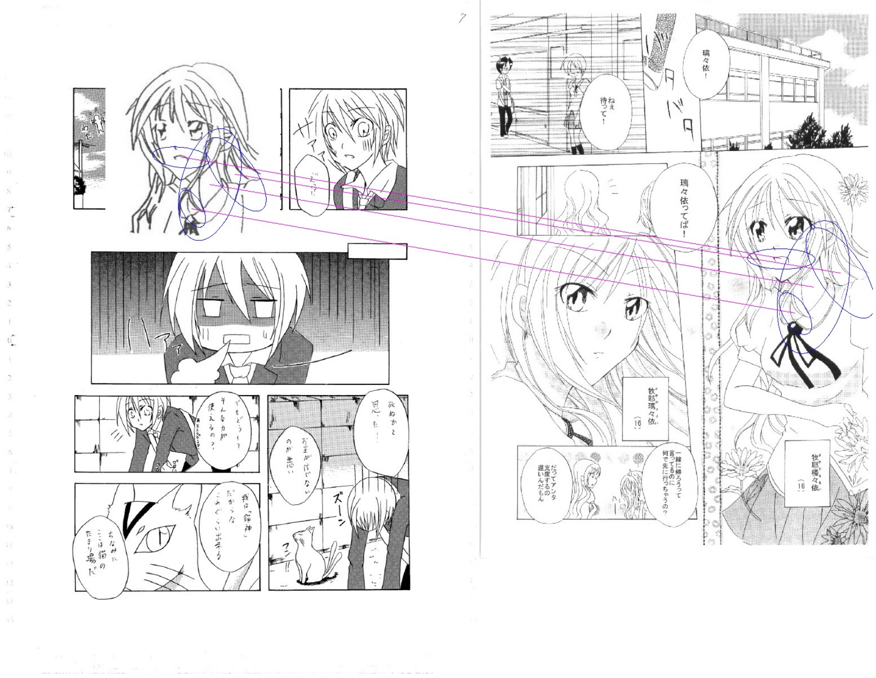
(b) Hand-drawn partial copy.

Figure 2.12: Experimental results of detecting partial copies from unknown backgrounds (num. of database images was $10,009$).

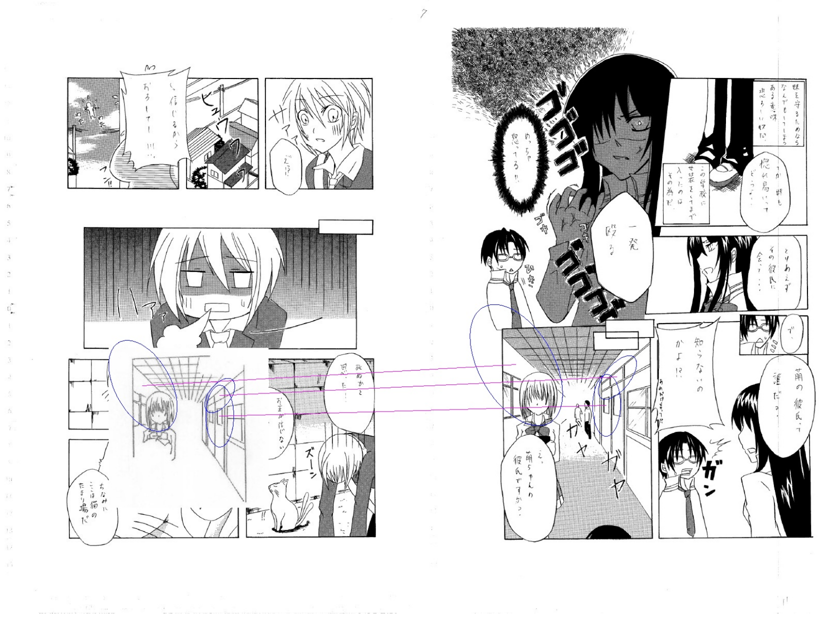
position information). The detection time was about 1.2 seconds per query.

2.4.6 Discussion

For an applicable method, we should consider three kinds of performance: database size, detection time and detection rate.

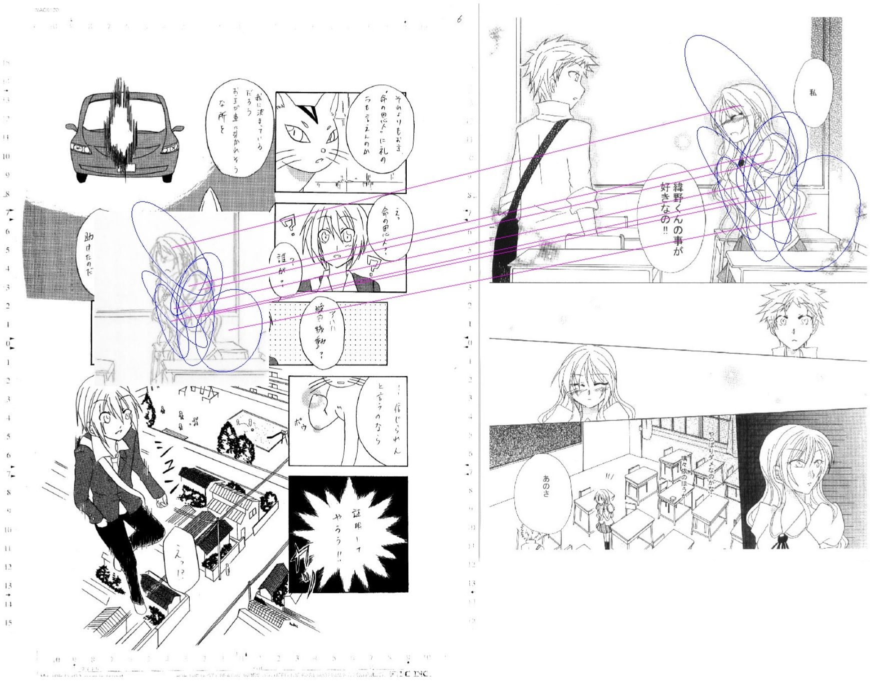


(a)

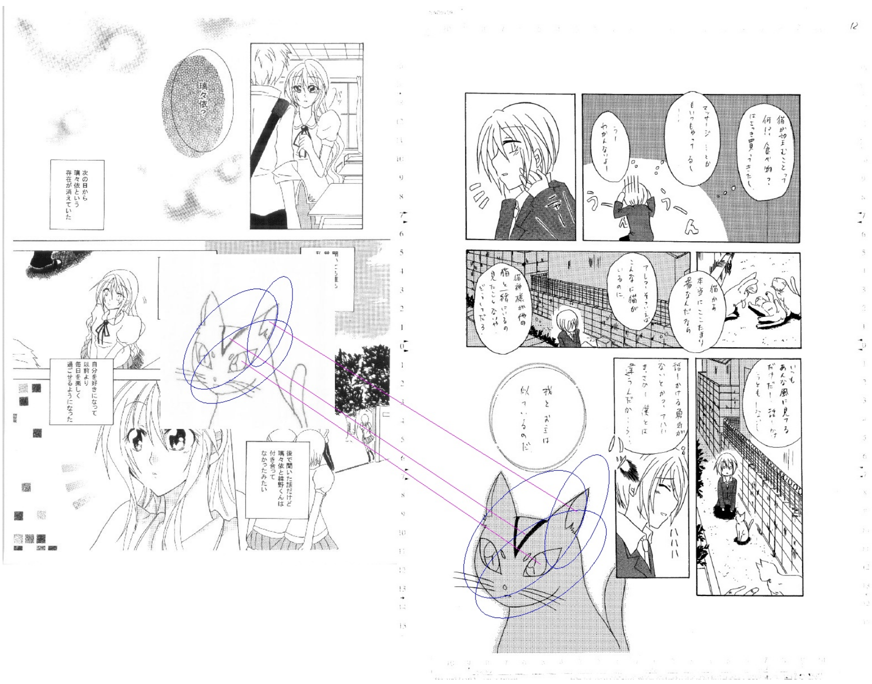


(b)

Figure 2.13: Examples of hand-drawn copy detection (Part 1). The left parts are query images with one hand-drawn partial copy inside for each. The right parts are the original one detected from a database of 10,009 manga pages. The ellipses represent LFRs, and the matched ROIs are connected with lines.

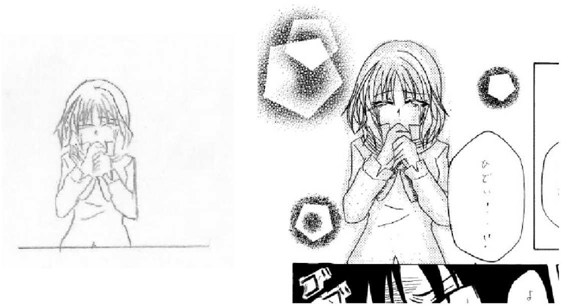


(a)



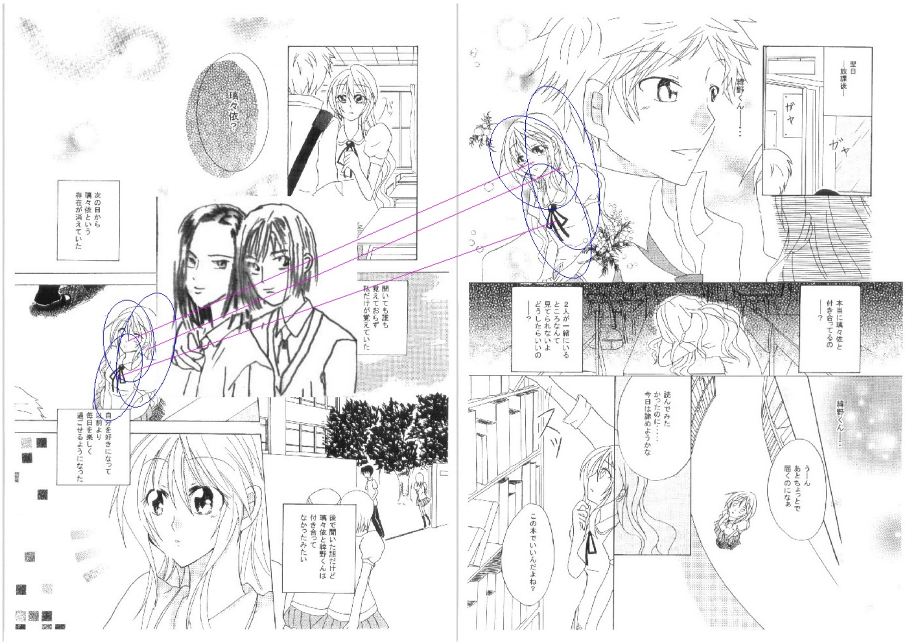
(b)

Figure 2.14: Examples of hand-drawn copy detection (Part 2). The left parts are query images with one hand-drawn partial copy inside for each. The right parts are the original one detected from a database of 10,009 manga pages. The ellipses represent LFRs, and the matched ROIs are connected with lines.



(a)

(b)



(c)

Figure 2.15: Examples of failure detections. The left parts are from queries, and the right parts are from originals. (a) shows an example of hand-drawn copy for which no results were retrieved. (b) shows an erroneous matching between hand-drawn copy and database images. (c) shows an erroneous matching between backgrounds and database images.

- Database size

  The database size depends on the series number of manga and how many pages per manga series. For example, if the system is for protecting 100 series of manga , and each manga series has three volumes (about 200 pages per volume), based on the experimental results we can see that the database for exact partial copy detection is about 20 GB. However, in practice, since the number of copyrighted manga is far larger than the assumption, methods for a more compact database is still a task for exact copy detection.

- Detection time

  The LFR detection time and feature description time depend on the contents and sizes of query manga pages. The matching time corresponding to each LFR will increase with a larger database. Although the exact copy detection require time for feature extraction and matching them, it is still much more efficient compared to manual check.

- Detection rate

  For copy detection systems, recall means the power to find copies, and precision means how much they can reduce manual work. For the proposed method, users can set parameters based on their requirements of recall and precision. In our experiments, we took about 50% as an acceptable precision, which means if users check two results manually, one illegal copy can be found. In practice, depending on the cases of users, the proposed method can offer a higher recall by reducing the precision. In addition, a recall above 75% of the proposed method could serve as deterrent to illegal users. From this point, we can see the contribution of the proposed method for the copyright protection of manga.

In addition, for both building the databases and detection of copies, manga pages are used directly without selection, cropping or any supervision in our system. This is a practical feature for actual use.

2.5 Conclusion

In this section, we focus on exact partial copies and propose to detect them by a image retrieval method using local feature matching. Comparing to the ineffectiveness of the previous method for the detection of hand-drawn copies, the proposed method achieve detecting both printed and hand-drawn copies from complex backgrounds. In addition, we propose a database construction method and achieve a better performance with a more compact database. By experiments, parameters of the proposed method was discussed and several approximate nearest neighbor searching methods were compared. Finally, we also test the performance of the proposed method based on a database of $10,009$ manga pages.

The future work includes increasing the detection speed and make the proposed method scalable for a larger database.

Chapter 3

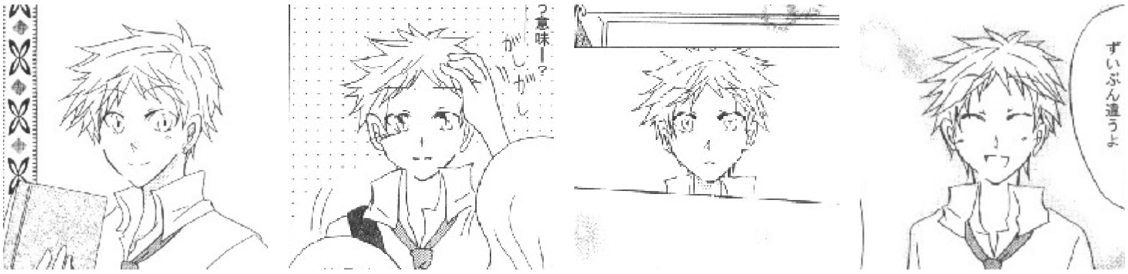
Similar Cartoon Character Detection Based on Region of Interest

3.1 Introduction

In chapter 2, we proposed using local feature matching to detect exact copies, which are based on specific parts of certain images. Besides exact copies, because of the simple compositions and abstract expressions applied in manga, similar copies can also be easily created. They always occur in infringing copyrights of main cartoon characters. By imitating their features, new drawings of the same cartoon character are created with different expressions. Since similar copies are not created based on the certain contents of specific manga pages, the method described in chapter 2 is not available for their detection.

For comic books, cartoon characters are an essential part and usually treated as symbols of the comics. The main characters appear throughout the whole series of comics, and act as cues of the story unfolds. Therefore, they are always the objects for similar copies and created by copying their features. As shown in Fig. 3.1, although the poses and facial expressions are different, they represent the same character. For the copyright protection of manga, similar copies of cartoon characters is a problem we have to consider.

In this chapter, we focus on the similar copy detection of cartoon characters. For this purpose, we propose to detect cartoon characters from manga pages by an object detection method and recognize them by feature matching. As the results, the proposed will report the location of the copies in manga pages, while offering similar characters as evidence. In the experiments, manga pages of 20 series are utilized to build the database. The cartoon



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

Figure 3.1: Examples of similar cartoon characters.

character of the same series but from different volumes are treated as similar copies to test the effectiveness of the proposed method.

As essential parts of cartoon characters, face regions are treated as Regions of Interest (ROIs) and utilized as the key for their detection and recognition. By our previous research [66], the method of human face detection [67] has been proved to be available for detecting cartoon character. In [68], we proposed recognizing the similar cartoon character by feature matching based on detected face regions Furthermore, we made a progress on the part of recognition and proposed Concentric Multi-Region (CMR) model to obtain significant features outside face regions [69]. In [70], we discussed the parameters of CMR model for performance and proved the scalability of the proposed method.

The rest parts of this chapter are arranged as follows: Section 3.2 is the related work. Section 3.3 introduces the proposed method. Experimental results and the discussions are shown in section 3.4. Finally, section 3.5 is the conclusion of this chapter.

3.2 Related Work

For human identification, methods of face detection and recognition are popular studied for decades, and offer us much related work.

3.2.1 Face Detection

Face detection is concerned with finding whether there are any faces in a given image and returning the location of each face, if present. This is the first step of any systems that analyze the information contained in faces.

According to [71], the face detection methods are classified into four categories: (1)

knowledge-based methods [72], which encode human knowledge of what constitutes a typical face, (2) feature invariant approaches [73], which find structural features invariant for changes caused by poses, viewpoints or lighting conditions, (3) template matching methods [74], in which several standard patterns of a face are stored to describe the face as facial features. and (4) appearance-based methods [67], whose models are learned from a set of training images to capture the representative features of facial appearance.

Considering the case of cartoon character faces, since they contain more changes than human faces without certain templates or prior knowledge, appearance-based methods are treated as the solution for their detection. However, the detection is still challenged by variations caused by facial features, occlusions and so on. To obtain representative properties of faces, numerous image representations have been proposed including pixel-based [75], parts-based [76], local edge features [77], wavelet feature [78]. Compared to these methods, the resent systems with Haar-like features [67],[79] showed impressive results in detecting faces under occlusion. In addition, with a greedy optimization algorithm to build the mode and the efficient feature computation, the method [67] demonstrated a real-time face detection. In our research [66], we also proved its effectiveness on cartoon character detection.

3.2.2 Face Recognition

Face recognition is concerned with determining the identity of a face in the image from a set of known labels. The technique are widely studied [80]–[82] and applied for human identification. These methods are only based on the features extracted from face regions and obtained good results.

Unlike human being, since manga are drawn with lines, the faces of cartoon characters are simple and without enough identifiable information. In contrast to this, not only the faces but also other parts like hear style and decorations around offer discriminative features for cartoon characters. In [69], we proposed using the features besides faces and achieved a better performance.

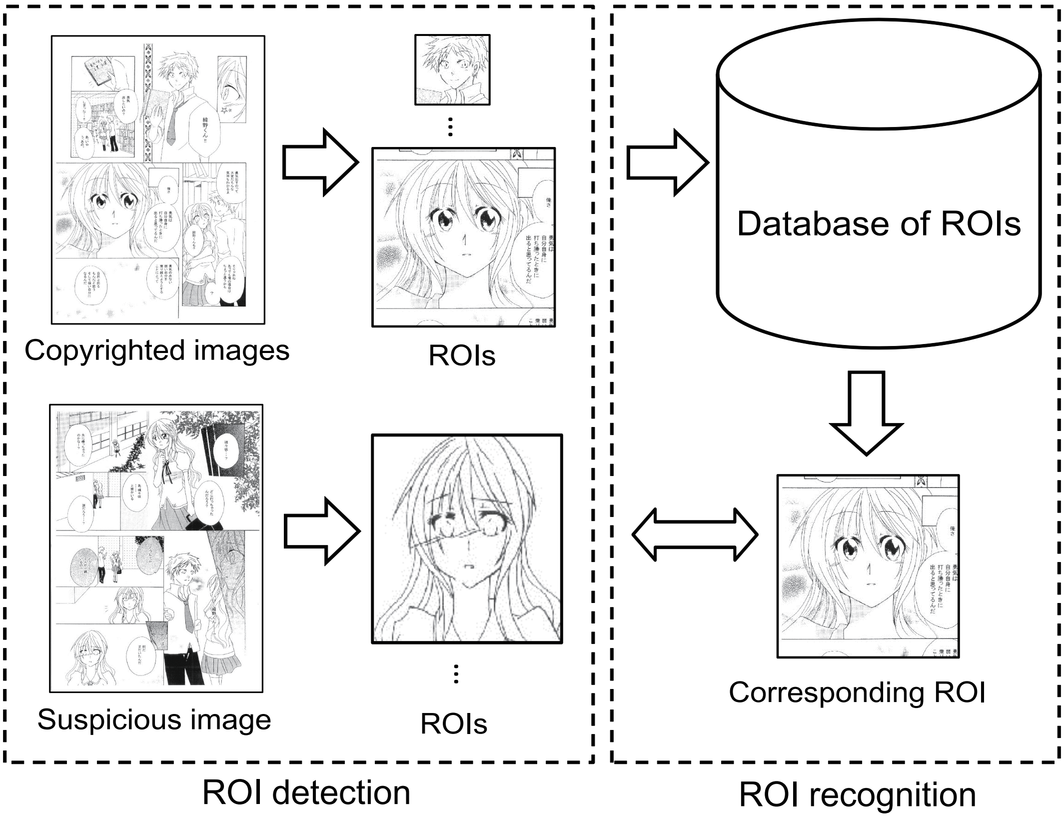
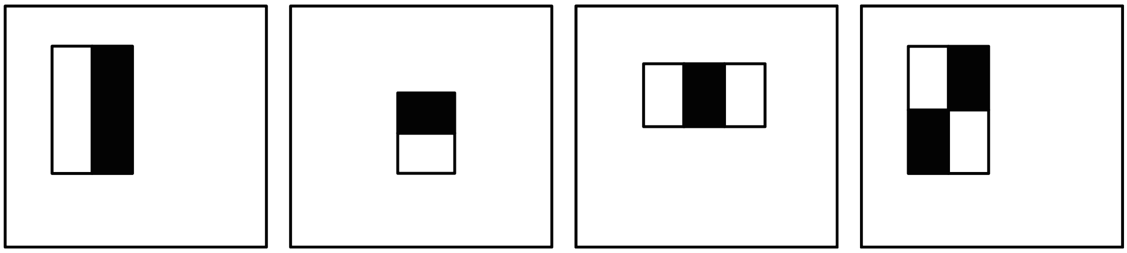


Figure 3.2: Processing of the proposed method.

3.3 Proposed Method

3.3.1 Outline of the Approach

Since the face regions are an essential part for cartoon characters, they are treated as ROIs in our method. As shown in Fig. 3.2, the processing of the proposed method is divided into two parts: ROI detection and ROI recognition. By ROI detection, ROIs are extracted from manga pages. In the part of similar ROI recognition, the ROIs extracted from copyrighted images are collected to build a database. The ROIs extracted from suspicious images are treated as queries and matched with ROIs in the database. If there is any matched ROIs, the query will be reported as a similar copy with its similar ROIs from the database as evidence.



(a) A. (b) B. (c) C. (d) D.

Figure 3.3: Example of using Haar-like features.

3.3.2 ROI Detection

To detect faces of cartoon characters, we propose applying Viola-Jones detection framework [67]. In this framework, first, a cascade classifier is trained, which is able to classify whether the object is in the detecting region (called detector). Then, the objects are detected from images by using the classifier.

For trainning the classifier, positive samples (images contain the object) and negative samples (non-object images) are collected. From each sample, features are extracted and marked with their response (1: positive, 0: negative). Based on a threshold for each feature, a decision tree (called weak classifier) can be created, which can predict the response of the detector. Then, the weak classifiers are selected based on their performances of predictions, and constructed to a cascade classifier.

3.3.2.1 Feature Extraction

Four kinds of Haar-like features are applied for describing the samples. As show in Fig. 3.3, the Haar-like feature is represented by the difference of sums of pixel values within adjacent rectangles. Since the size of rectangles changes from 1 pixel to the whole detecting region, there are $162,336$ features for a region with the size of 24×24 pixels.

To speed up the calculation of Haar-like features, integral image is applied, by which the sum of pixels for any region can be obtained by 3 times' calculations. Therefore, the Haar-like features can be calculated in a constant time.

Besides Haar-like features, other features can also be applied to the Viola-Jones framework. A more precise description of objects' characteristics can lead to a better performance.

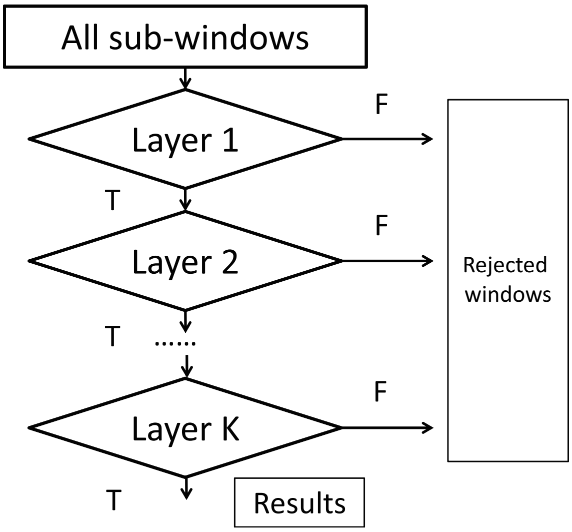


Figure 3.4: Detection using cascade classifiers.

3.3.2.2 Feature Selection

The Haar-like features are over-complete, which hamper the fast and robust detection. In the framework, Adaboost [30] is utilized to select the most effective ones. AdaBoost is a machine learning method to combine weak classifiers into a strong classifier by an iterative algorithm. In each loop, a weak classifier, which is the decision tree built by corresponding Haar-like feature, with the lowest error rate is generated. By setting the highest error rate, some high-performance weak classifiers are retained to the next step.

3.3.2.3 Construction of Cascade Classifier

Furthermore, as shown in Fig. 3.4, a cascade structure for arrangement of weak classifiers is imported to increase the detection performance while reducing computation time.

Given the detection rate and false positive rate are d_i and f_i for layer i, respectively. The detection rate (D) and false positive rate (F) of final cascade classifier will be

$$D = \prod_{i=1}^{K} d_i \ , \quad F = \prod_{i=1}^{K} f_i$$

where K is the number of cascade layers. By reducing the threshold of classifiers, it is easy to achieve a high detection rate, such as $d_i = 0.995$, with a high false positive rate, such as $f_i = 0.5$ for each layer. However, with increasing K, the false positive rate D is reducing much faster than the detection rate D of the cascade classifier. In addition, since

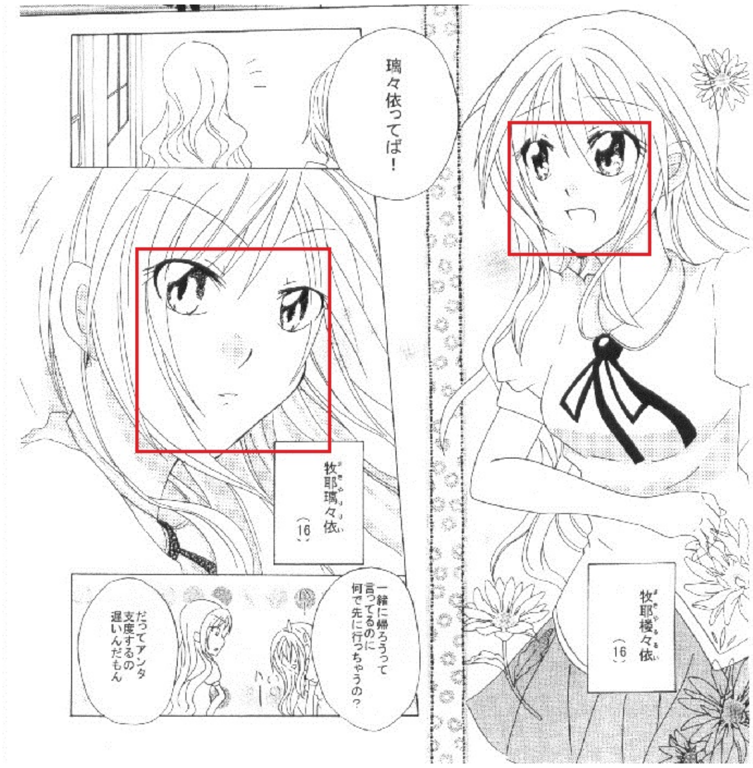


Figure 3.5: Example of ROI detection.

the rejected patterns (sub-windows) are not classified in the next layers, the detection also becomes faster.

Besides the setting of d_i, f_i and K, the error rate is also affected by the number of positive samples and negative samples. We will discuss that by experiments in the section 3.4.1.

3.3.2.4 Detection of ROIs

The sliding window technique is applied to detect sub-windows in different scale. After detecting all sub-windows of the whole image, the union-find algorithm is applied to group detected sub-windows. As shown in Fig. 3.5, square face regions R_i ($\{x_i, y_i, a_i\}$) are detected from the comic page, where (x_i, y_i) is the center, a_i is the side length of R_i. By this method, we can locate the cartoon characters in the manga pages and obtain their face regions as unit ROIs, which are from just above the eyes to chins of cartoon characters.

3.3.3 ROI Recognition

After ROI detection, the ROIs from copyrighted manga pages are collected for a database. The characters from suspicious manga pages are treated as queries and recognized: described and matched with similar ones in the database. For the recognition, the discriminative feature presentation is important. Based on the ideas that (1) face regions are the most significant parts and (2) other identification features are around face regions, we proposed CMR model to obtain the discriminative features of cartoon characters.

3.3.3.1 Concentric Multi-Region Model

Normally, the faces are in the centers of these detected regions, as shown in Fig. 3.5. Since detected regions converge to some general features of faces, such as two eyes, one mouth and so on, it is difficult to classify cartoon characters only considering the information extracted from detected regions. Therefore, we enlarge the range of detected regions as ROIs. With the increased size of ROIs, we can obtain more discriminative information. As shown in Fig. 3.6, if only considering face regions (from eyebrows to chins), we may have the answer of Fig. 3.6(b) as the similar matching with Fig. 3.6(a) since they have the similar face expressions. With regard to the features in a larger range, such as hair and collars, Fig. 3.6(c) can be reported, which is the right answer, since it belongs to the same character with Fig. 3.6(a). Whereas noises are also increased in larger ROIs. For example, as shown in Fig. 3.7, although Fig. 3.7(a) and Fig. 3.7(b) express the faces of the same character, the character around the main character of Fig. 3.7(b) covers some parts and decreases its similarities to Fig. 3.7(a). In contrast to it, Fig. 3.7(c) is more similar to Fig. 3.7(a) for the similar contours of the characters.

Another problem is resolutions for descriptions. Some stable features should be described in detail, and unstable ones should be described coarsely. Usually the features near the centers of the objects are more stable than the ones far from the centers. For example, as shown in Fig. 3.7, the glasses of the character in Fig. 3.7(a) is the key to classify them. Although the glasses is much smaller than the character around in Fig. 3.7(b), it is more discriminative. Therefore, the regions near the center are stable and need a description with higher resolution, and the regions far from the center are not stable but should be also considered in a low-resolution way.

Based on the ideas above, we propose a Concentric Multi-Region model to describe the object for the recognition, as shown in Fig. 3.8. It has a pyramid structure, in which layers

(a) (b) (c)

Figure 3.6: Examples of similar patterns of cartoon characters. Fig. (a) is similar to Fig. (b) for the small region around the center. Fig. (a) is similar to Fig. (c) for the large region around center. Fig. (a) and Fig. (c) to the same character.

(a) (b) (c)

Figure 3.7: Examples of similar pattern of cartoon characters. Fig. (a) is similar to Fig. (b) for the small region around the center. Fig. (a) is similar to Fig. (c) for the large region around center. Fig. (a) and Fig. (b) to the same character.

represents ROIs sharing the same center. The size of ROIs are increasing and described coarsely from the top to the bottom. Instead of searching the most discriminative ROI among them, we search the most discriminative features f_n^i from each layer n and combine them into a feature vector $\boldsymbol{F} = \{f_1^1, f_1^2, ..., f_n^l ...\}$ as the description of the region. The detected region is treated as a unit ROI ($R_0 = \{x, y, a\}$) which is the top layer of the pyramid. The ROI of layer n is set to $R_n = \{x, y, a \times (1 + n/8)\}$ which shares the same center (x, y) of the unit ROI, but with a larger size. In the section 3.4.2, we discussed the parameter n for recognition.

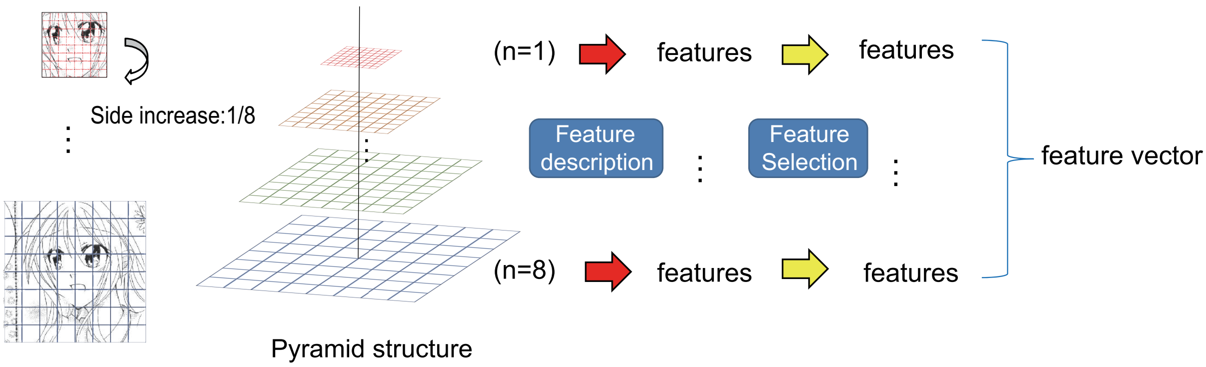


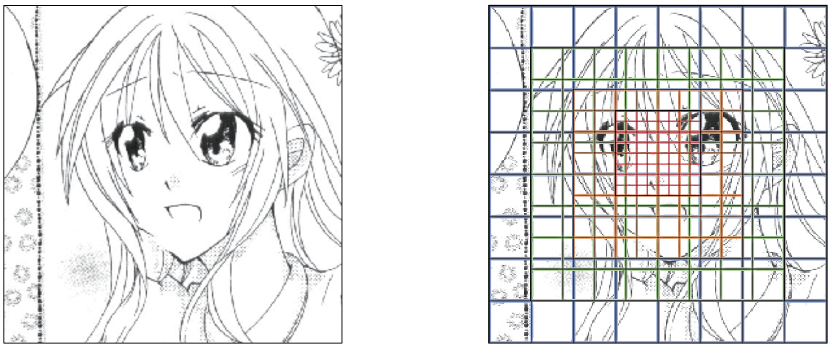
Figure 3.8: Processing of Concentric Multi-Region description.

3.3.3.2 Feature descriptor

To describe each layer of CMR, we apply HOG (Histograms of Oriented Gradients) [29]. It was proposed for human detection, but also extended to recognition areas, such as face recognition [81] and handwritten character recognition [83]. Rather than using precise edge positions, HOG describe objects by the distribution of their edge directions. Because lines of similar characters are not exactly same, HOG would be a good choice for their description.

The specific calculation of HOG is described in section 2.3.3. The parameter setting of HOG is based on our preliminary experiments of cartoon character recognition. For each ROI, there is 8×8 cells and 6×6 blocks (3×3 cells per block). The gradient orientation is quantized into 6 bins. From the discussion of [29], fine orientation coding performed better, but in our experiments, there is no significant difference between 6 and 9 bins for orientation. As a results, we obtain one $1,944$ dimensional HOG feature vector for each ROI.

Although the parameters of the feature descriptor are the same for different layers in the Concentric Multi-Regions Model, the fineness of descriptions are different because the sizes of ROIs are different. As shown in Fig. 3.9, compared to cells in bottom layers of the pyramid, cells in the top layers (parts closer to the center) contain less pixels which cause a finer description.



(a) description region. (b) different resolutions.

Figure 3.9: Concentric Multi-Region description (Higher resolutions for center parts).

3.3.3.3 Feature selection

For each layer, we select the most identifiable features by the AdaBoost algorithm [30]. AdaBoost is a machine learning method. Given a data set of positive samples and negative samples, AdaBoost can be used to learn a binary classification function (strong classifier) combined with weak classifiers by an iterative process.

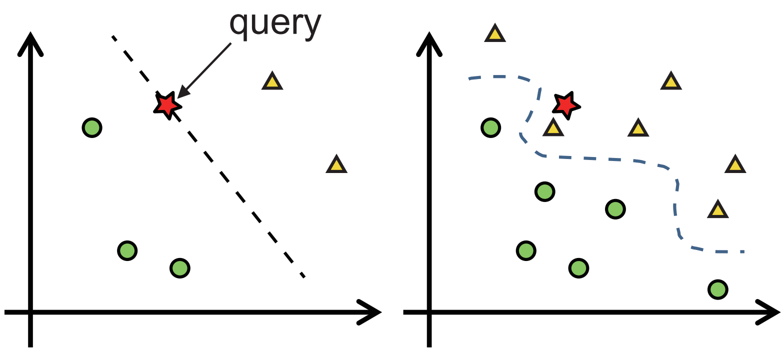
As shown in Fig. 3.10, first, we collected S images of K cartoon characters (we applied 10 characters and 100 patterns for each in this research) as samples which are described into HOG features and labeled with their IDs ($k \in [1, K]$). For each sample, we add an extra feature e which is assigned from 1 to K, and if e equals to its ID, it is treated as a positive sample, otherwise not. Thus, we can obtain $N = K \times S$ samples ($\boldsymbol{x}_i$,y_i), where $\boldsymbol{x}_i$ is the features (HOG features plus one extra feature e) and y_i is the response for sample i. At the beginning, the same weight W_1 is assigned to samples. For each feature in $\boldsymbol{x}$, a decision tree is built as a weak classifier $h(\boldsymbol{x})$, which can predict the results (1: positive, 0: negative) by features of samples. Through the predictions of all samples in each iteration t, the weak classifier $h_t(\boldsymbol{x})$ which has the lowest error is selected, and weights of samples are updated to W_{t+1} for the next round. The weights of the samples failed to be predicted will be set greater to make the next weak classifier emphasis on them. If the error rate equals to 0.5, it means no classifier can offer a better classification than random guessing, thus the algorithm break the iterations and combine the selected weak classifiers with their reliability into a strong classifier. After training, we remove the weak classifier of the extra feature e and sort the others by their reliabilities. The top M features of each layer of ROI are utilized and connected as the final feature vector named CMR-HOG feature vector.

1. Given N samples $(\boldsymbol{x}_1,y_1)$,...,$(\boldsymbol{x}_N,\ y_n)$ where $\boldsymbol{x}_i$ is feature description of sample i, y_i is response: 0 for negative samples and 1 for positive samples.
2. Initialize weights $W_1(i) = \frac{1}{N}$ for each sample.
3. For t = 1,...,T:
   (a) Select the best weak classifier $h_t(\boldsymbol{x})$ with respect to the weighted error
   $$\epsilon_t = \sum_{i:y_i \neq h_t(\boldsymbol{x}_i)} W_t(i)$$
   (b) Calculate reliability α_t of h_t as
   $$\alpha_t = \frac{1}{2}\log(\frac{1-\epsilon_t}{\epsilon_t})$$
   (c) Update the weights for samples:
   $$W_{t+1}(i) = W_t(i)\exp[-\alpha_t y_i h_t(\boldsymbol{x_i})]$$
   For samples recognized correctly, weights become smaller as $W_{t+1}(i) = W_t(i)\exp(-\alpha_t)$.
   For samples failed to be recognized, weights become larger as $W_{t+1}(i) = W_t(i)\exp(\alpha_t)$.
   if $\epsilon_t = 0.5$ exit loop
   (d) Normalize the weights:
   $$W_{t+1} = \frac{W_{t+1}(i)}{\sum_{i=1}^{N} W_{t+1}(i)}$$
4. The final strong classifier is:
   $$H(\boldsymbol{x}) = \text{sign}[\sum_{t=1}^{T} a_t h_t(\boldsymbol{x})]$$

Figure 3.10: AdaBoost algorithm for feature secleciton.

3.3.3.4 Matching

The detected ROIs are matched based on their feature vectors. We calculate the Euclidean distance between the CMR-HOG feature vectors, and match the pairs which are



(a) (b)

Figure 3.11: Feature spaces of ROIs. (a) shows a feature space of ROIs with few patterns for each character. (b) shows a feature space of ROIs with more patterns for each character.

near each other. We applied ANN (Approximate Nearest Neighbor Searching) [50] to speed up the matching, whose detail is described in section 2.3.5. Since there are more than one patterns (ROIs detected from different manga pages) for each character in the database (detail is described in the next section), one ROI detected from queries can be matched with more than one patterns in the database. Denote $\boldsymbol{q}$ as the feature vector extracted from a ROI of the query image and $\boldsymbol{p}$ as the feature vector in the database, the matching should satisfy $d(\boldsymbol{q}, \boldsymbol{p}) < T_f$. ($T_f$ is a threshold), otherwise not.

3.3.3.5 Database construction

For the database, all ROIs detected from copyrighted manga pages are applied without any selection. In addition, since there are many variations in the faces of manga characters, such as facial expressions and poses, more possible patterns in the database can offer a better performance. As shown in Fig. 3.11, the query can be well recognized if more patterns of certain characters are applied. The implementation is to simply increase manga pages for building the database. We show the relation between database size and recognition performance in the section 3.4.2.

3.4 Experimental Results

In this section, we test the effectiveness of the proposed method by using real manga publications as data. All the experiments were done using a computer with CPU INTEL Core i7-870 2.93 GHz and RAM 8 GB. The detection results are reported by using recall $R = A/B$ and precision $P = A/C$, where A is the number of correct detected patterns, B is the number of correct answers, and C is the number of detected patterns. Experiments are divided into two parts: (1) ROI detection, which is to detect cartoon characters from manga pages, and (2) ROI recognition, which is to recognize the detected cartoon characters. Thus, the definitions of correct answers in the two experiments are different: faces for ROI detection and the same character for ROI recognition. Finally, we make a discussion of similar cartoon character detection by using the proposed method based on the results of these two experiments.

3.4.1 ROI Detection

First, we did an experiment to test the performance of ROI detection. The parameters of cascade classifiers were set as number of layers $K = 20$, detection rate $d_i = 0.995$ and false positive rate $f_i = 0.5$ for each layer. For the training set, we collected faces of cartoon characters from 15 series of manga as positive samples. As shown in Fig. 3.12, most of the faces are cropped from just above the eyebrows to chin, and normalized to 24×24 pixels as our positive samples. The patterns which do not contain any face patterns were applied as negative samples. For the validate set, 101 images (about 800×1200) were chosen from 21 series outside the training set. In the validate set, there are 365 faces, which are treated as right answers.

The detection results are shown in Table 3.1 by different classifiers trained with different data set. The results show us the trade-off of between precision and recall. Take the classifier C1 as a benchmark, we can see that a larger number of negative samples can increase precision and decrease recall. In contrast, increasing positive samples can lead a higher recall with lower precision. By increasing both positive and negative samples, we can obtain a lower error rate of both precision and recall.

The classifier C4 achieved a relative high performance and was applied for the rest of experiments. The examples of detection are shown in Fig. 3.13. We can see that although the faces of cartoon characters contain many changes comparing with real faces of human beings, we have achieved the accurate detection.

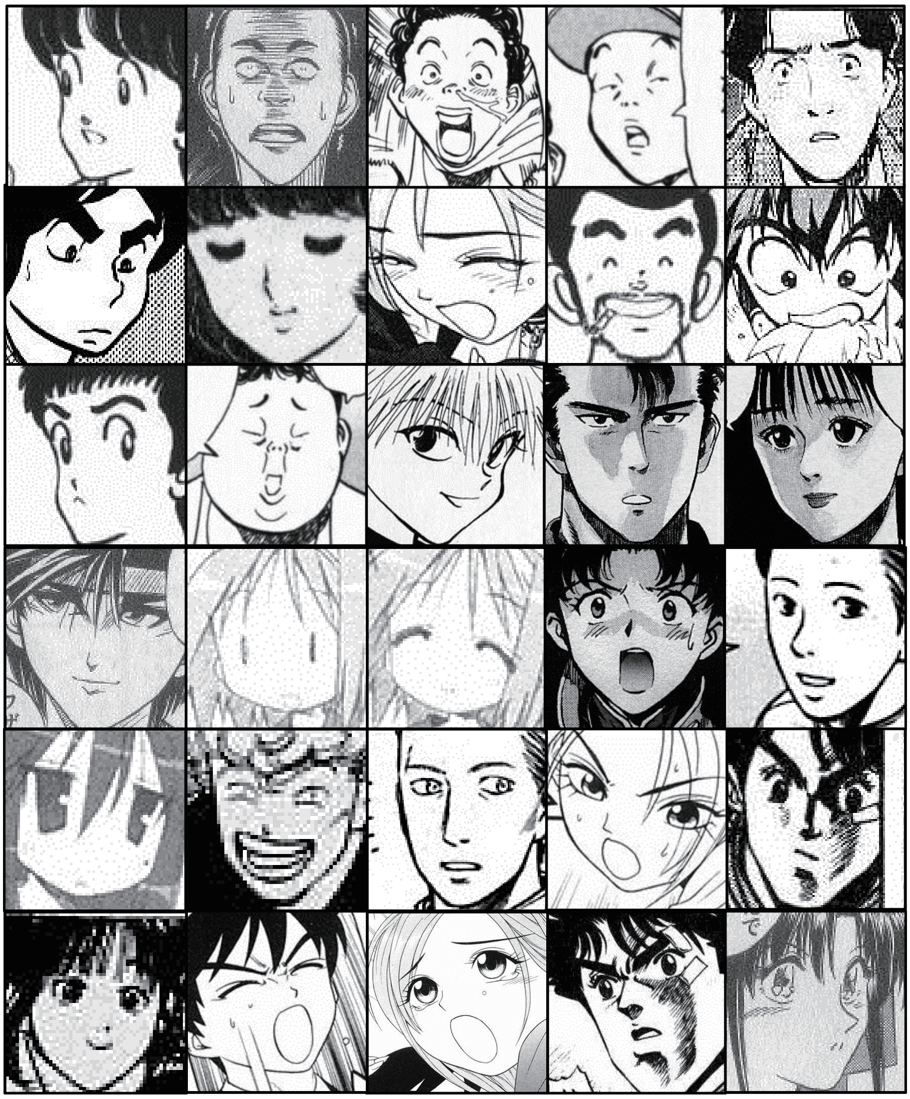


Figure 3.12: Examples of positive samples used for training. They are from Rurouni Kenshin, Neon Genesis Evangelion, Hoshin Engi, H2, Hunter $\times$ Hunter, JoJo's Bizarre Adventure, Lucky Star, Master Keaton, Maison Ikkoku, Miyuki, Monster, Planetes, Rosario + Vampire, Rough and Slam Dunk.

As shown in Fig. 3.14, we calculated the mean image of training set. We can recognize the shape of face from the mean image of positive samples, but not from the mean image of negative samples, from which we can see there are still some discriminative features in faces of cartoon characters.

The ROI detection time depends on the size of query images and the number of ROIs in one image. In our experiments, the average detection time (classifier C4) is 472 ms per

Table 3.1: ROI detection results by different classifiers trained with different numbers of training samples.

Classifier	C1	C2	C3	C4
No. of positive samples	$1,000$	$1,000$	$3,000$	$3,000$
No. of negative samples	$1,000$	$3,000$	$1,000$	$8,000$
Precision (%)	65.0	92.5	33.1	87.0
Recall (%)	80.8	60.6	91.0	80.3

manga page (142 ms per ROI).

3.4.2 Face Recognition

Then, we test the effectiveness of the proposed method for the recognition of cartoon characters.

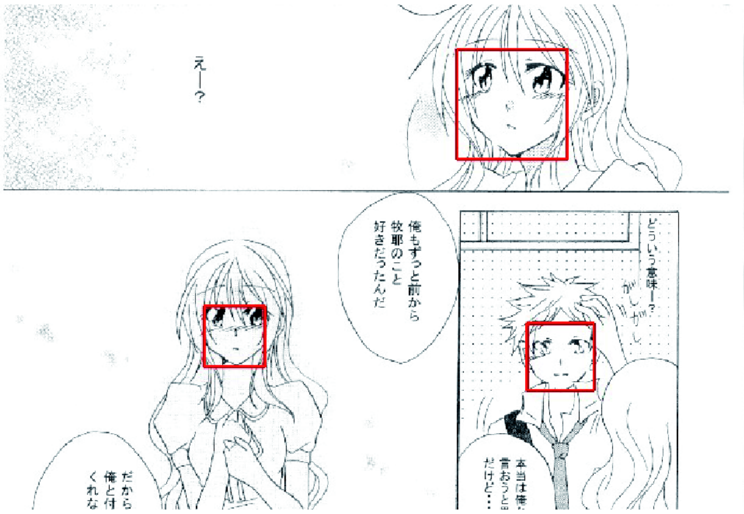
3.4.2.1 Experimental Setting

To make a clear evaluation of the performance on the recognition, we utilized labeled pattern of cartoon characters, which were detected by the ROI detection. From 20 series of manga, we chose 27 cartoon characters and labeled all their patterns with their IDs. As shown in Fig. 3.15, the patterns belong to the same character, which are drawn in different manga pages, are not exactly same. We treated the patterns with same label as similar ones and matched them with other patterns. Only the matched patterns with the same label to queries were treated as correct answers for recognition. The parameter of approximation in ANN is set to $\varepsilon = 5$.

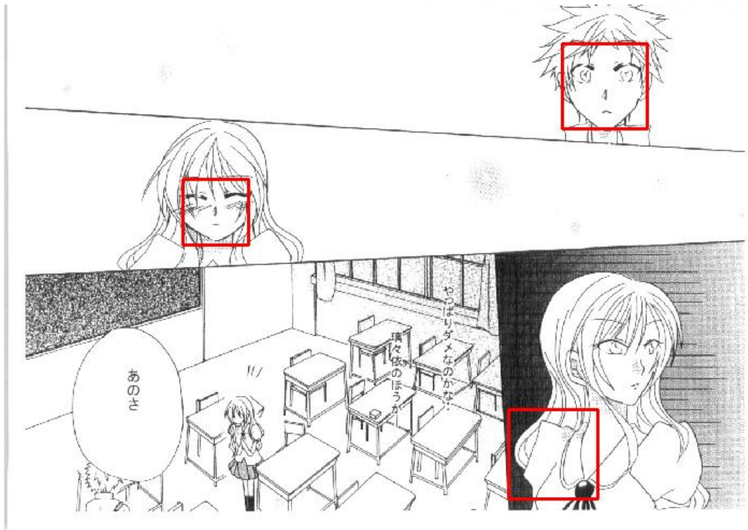
3.4.2.2 Experiment 1

First, we trained CMR-HOG descriptor and discussed the number of layers n in CMR model by experiments.

As the training dataset, $1,000$ face images from 10 labeled characters (100 images for one characters) are applied. With an extra feature, $1,000$ positive samples and $9,000$ negative samples are created. By using these samples, we trained the strong classifiers



(a)

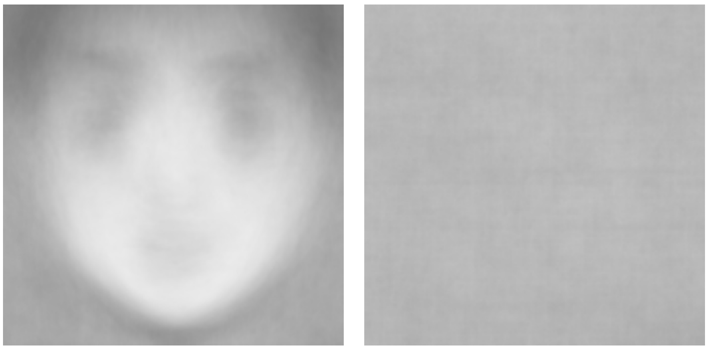


(b)

Figure 3.13: Examples of face detection of cartoon character. . There is a false positive and a false negative at the bottom right of Fig. 3.13(b) (detector was trained with 3, 000 positive samples and 8, 000 negative samples).

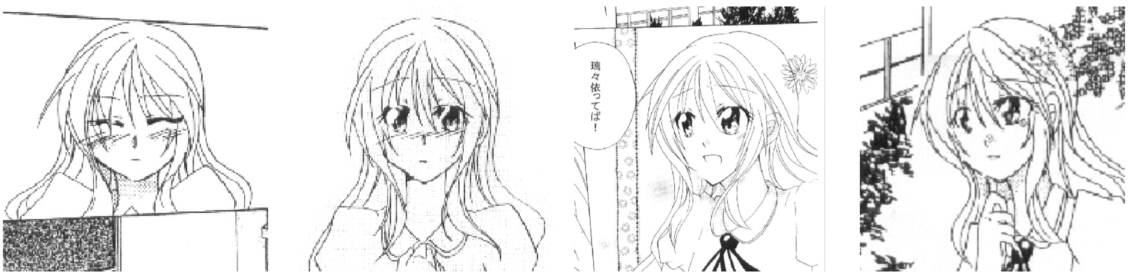
and obtained about 150 weak classifiers for each layers. 140 features with high reliabilities are chosen from each layer, therefore, we obtained CMR-HOG feature vectors with $140 \times n$ (n is the layers applied in CMR model) dimensions.

As the validation dataset, we applied 1, 377 patterns of the same 10 characters but outside the training dataset. By changing n from 0 to 11, we obtained the results as shown in Fig. **??**. The increase of n led a better performance. Since there was no



(a) Mean image of positives. (b) Mean image of negatives.

Figure 3.14: Mean images of training samples.



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

Figure 3.15: Examples of patterns of the same cartoon character from different manga pages.

significant improvement for n larger than 8, we chose $n = 8$ for CMR model and applied the descriptor for the rest of experiments. Therefore, the applied CMR-HOG feature vectors are $1,120$-dimensional.

3.4.2.3 Experiment 2

Then, we compared the effectiveness of CMR-HOG with other descriptions based on face regions.

In this experiment, we built a database containing $10,827$ patterns detected from 20 titles of comics[2]. In the database, there were 17 labeled characters outside the training dataset used in section 3.4.2.2. Among them, there are 500 patterns for 10 labeled characters (50 for each character) and 75 patterns for the rest 7 characters. The 575 labeled patterns were treated as queries and matched with other patterns (except themselves) in

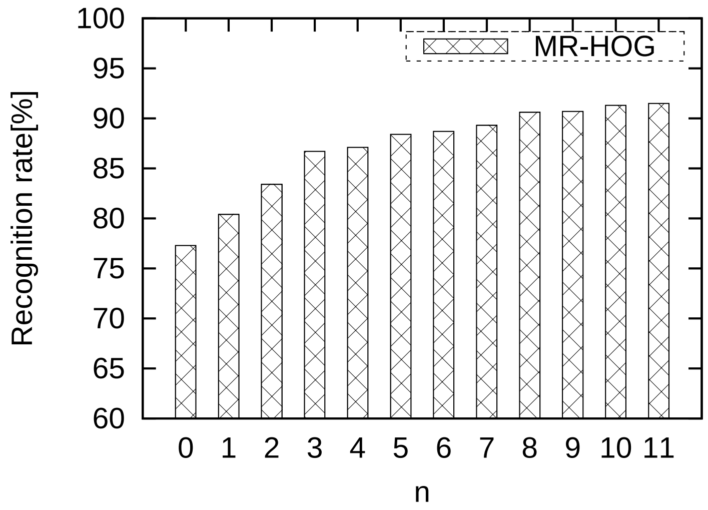


Figure 3.16: Results for CMR-HOG with different number of layers n.

the database. For each query, only the nearest one in the database was reported.

The experimental setting was the same as the first experiment. As methods for comparison, three methods based on face regions are applied. Their recognition method are the same as the proposed method but with different description methods: (1) HOG, (2) GW (selected Gabor Wavelet features) [80] and (3) PHOG (Pyramid Histogram of Oriented Gradients) [84] described below.

HOG The parameter setting is the same as section 3.3.3.2. For each region, a HOG feature vector of $1,944$ dimension was extracted.

GW Two-dimensional Gabor wavelets was introduced by Daugman [85] for human iris recognition, and widely applied for face recognition [86],[80],[82]. We followed the way in [80] to select discriminative features from a large set of Gabor wavelet features. First, we scaled the face regions to a fixed grid size (20×20), and convolved them with 40 Gabor wavelets (4 scales and 8 orientations of the Gabor kernel). $16,000$ Gabor wavelet features were obtained from each region. By the AdaBoost algorithm, we selected 350 discriminative features based on the same training dataset applied in the proposed method. The feature vector joined with selected features was applied for the recognition.

PHOG It is a spatial pyramid extension of HOG descriptor and have shown good

[2]The comics applied in experiments include “20th Century Boys”, “Kare Kano”, “Neon Genesis Evangelion”, “Hoshin Engi”, “H2”, “Hunter × Hunter”, “Stone Ocean”, “Lucky Star”, “Master Keaton”, “Maison Ikkoku”, “Miyuki”, “Monster”, “Rozen Maiden”, “Planetes”, “Rosario + Vampire”, , “SLAM DUNK” “Rurouni Kenshin”, and 3 comics drawn for this research.

Table 3.2: Recognition results using different features (HOG, GW, PHOG and CMR-HOG).

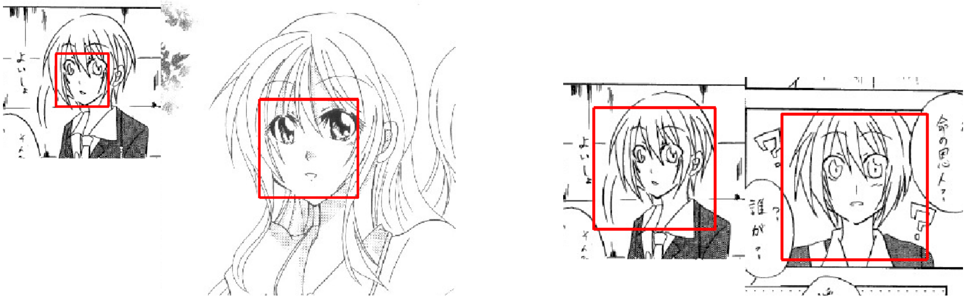
Feature description	Recognition rate
GW	47.8%
PHOG	59.7%
HOG	69.5%
CMR-HOG	83.6%

performances in object recognition [84] and facial expression recognition [87]. In PHOG, the described region is divided into spatial grids at all pyramid levels, and the resolutions of grids are different for different levels. The gradients divided into bins of each grid are joined together to be the description of the region. Based on [87], we set the number of pyramids to 3, the bin size to 8 at the orientation range of $[0, 360)$ degrees. We obtained a PHOG feature vector with 680 dimension for each region.

The experimental results for the recognition of cartoon characters are shown in Table 3.2. From the results, we can see that although the methods of HOG, GW and PHOG have shown good performances in face recognition and other object recognition, for cartoon characters, they lose the effectiveness. This is because the comics only employ simple lines to represent characters, and thus, there are many similar faces for different characters. As shown in Fig. 3.17(a), it is difficult to classify the two characters just based on their face regions. Whereas other parts like hair style and decorations are applied as their identification features. By employing the discrimination information around the face regions, the proposed method performed much better than the other three methods. As shown in Fig. 3.17(b), the characters share simlar face features but different hair style, by which the proposed method achieved the recognition.

Some examples of successful recognition by the proposed method are shown in Fig. 3.18. Only the regions inside the bonding boxes were used for description. We can see that our method achieved the recognition of cartoon characters. Even if the query pattern contains some noises as shown in Fig. 3.18(d), facial expression changes as shown in Fig. 3.18(e) and different poses as shown in Fig. 3.18(f), they can also be correctly matched with the similar ones in the database.

The main reasons for failures are threefold. (1) Serious noises and changes. Such as shown in Fig. 3.19(a), the characters are closing their eyes while covering her mouth with hand. (2) Strange facial expression. It is also an attractive point for comics, as shown



(a) (b)

Figure 3.17: Example of correct matching by the proposed method but failed by comparing methods. Fig. (a) shows an example of recognition result by the comparing method of HOG. Fig. (b) shows an example of recognition result for the same query by the proposed method. (The left side shows a query pattern and right side shows the matched pattern of the database. The regions inside bounding boxes are used for descriptions.

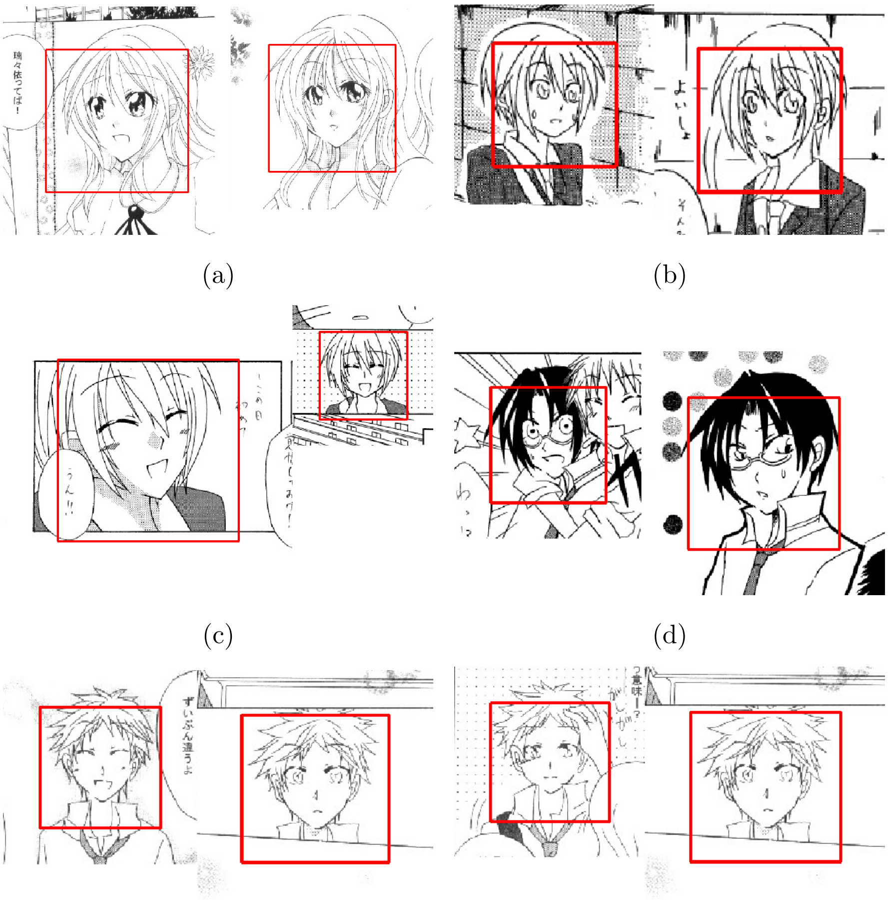
in Fig. 3.19(b). (3) Changes of regular features. Normally, cartoon characters keep their hair style or decorations as regular features for audiences' recognition. However, there are exceptions, such as shown in Fig. 3.19(c): the pattern which represents the same character as Fig. 3.7(a) has taken off his symbolic glasses.

3.4.2.4 Experiment 3

Next, we checked effects of volumes of patterns in the database for the proposed method. As the queries, we applied 500 patterns from the 10 labeled characters applied in section 3.4.2.3. Three databases were made for this experiments: DB1 was the same as section 3.4.2.3 ($10,827$ patterns from 20 titles of comics); DB2 contained all the patterns of DB1 plus additional 50 patterns for each labeled characters (100 patterns for each labeled character); For DB3, $11,535$ patterns from other 5 titles of comics[3] were added to DB2. In this experiment, the top 5 matched patterns are reported. The cumulative recognition rates are shown in Fig. 3.20.

Because of noises happened in comics and changes of cartoon characters, the detected patterns may be partly similar with a different character. Therefore, checking the similar patterns more than the nearest one can increase the recognition rate. Comparing to DB1, recognition with DB2 performed much better (3.6% increasing for recognition rate of the

[3]The comics include "Fighting Spirit", "Rough","Phantom Blood" and "Tomehane".



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

Figure 3.18: Examples of right matching. The left side shows a query pattern and right side shows the matched pattern of the database. The regions inside bounding boxes are used for descriptions.

(a) (b) (c)

Figure 3.19: Examples of patterns failed for recognition.

top rank). This is because more similar patterns increase the regular features of a certain character in the database. Although DB3 was more than twice as DB2, the recognition

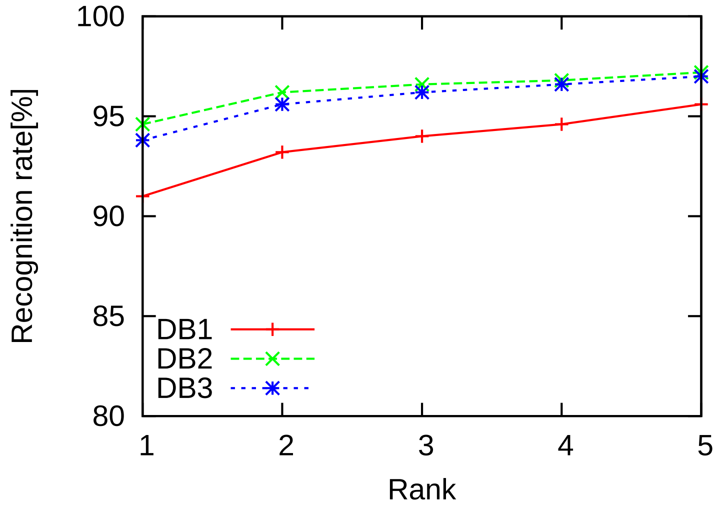


Figure 3.20: Results for different database.

rate did not change a lot, from which we can see the stability of the proposed method for character recognition. The detection time for DB1 and DB2 was 11 ms per pattern and 23 ms per pattern for DB3. Although the recognition requires more time with the increase of patterns in the database, it is still fast for all characters of 25 titles of comics.

3.4.2.5 Discussion

As the conditions of our experiments, if over 50 patterns for each character (it is not difficult for main characters of famous comics), our method can recognize more than 90% of detected patterns. By checking more candidates, for example, the top 5 similar patterns, the recognition rate can be over 95%.

In addition, since there are many cartoon characters in comic publications, the recognition based on a large database is required. Based on the experiments, we can see the fast and stable recognition of the proposed method, which are mandatory for practical use.

3.4.3 Overall performances

The proposed method contains two parts: ROI detection and ROI recognition. Therefore, the recall (R_s) and precision (P_s) of the whole system are $R_s = R_d \times A_r$, $P_s = P_d \times A_r$, in which R_d, P_d, A_r, represent recall and precision of ROI detection and recognition rate of cartoon character, respectively. In the experiments, we achieved $P_d = 87.0\%$ and

$R_d = 80.3\%$ using the classifier trained by $3,000$ positive samples and $8,000$ negative samples, and $A_r = 94.5\%$ for the copyrighted cartoon character which have 100 patterns in the database. Based on such a condition, we can obtain a performance as $P_s = 82.2\%$ and $R_s = 75.9\%$ for the all system. The detection time of the system also includes two parts of time: ROI detection time and similar ROI recognition time. For the same condition, the average detection time is 165 ms per ROI.

3.5 Conclusion

In this paper, we focus on similar copies of cartoon character and propose a method to detect them based on ROIs. The method consist of two parts: ROI detection and ROI recognition. As a basic part, face regions are treated as ROIs. The cartoon characters are located in the manga pages through ROI detection and recognized in ROI recognition. For the recognition, considering the identification information of cartoon characters are not limited in face regions, we proposed Concentric Multi-Regions Model to explore significant features outside face regions for increasing the recognition rate. In the experiments, we discussed the effect of training samples to the detection and the parameters of recognition model. From the experimental results, we proved the effectiveness of the proposed system for detecting similar copies of cartoon characters.

The future work includes

- increasing the recall and precision of ROI detection
- detecting other important parts of cartoon characters besides face regions
- applying a larger database to test the proposed method

Chapter 4

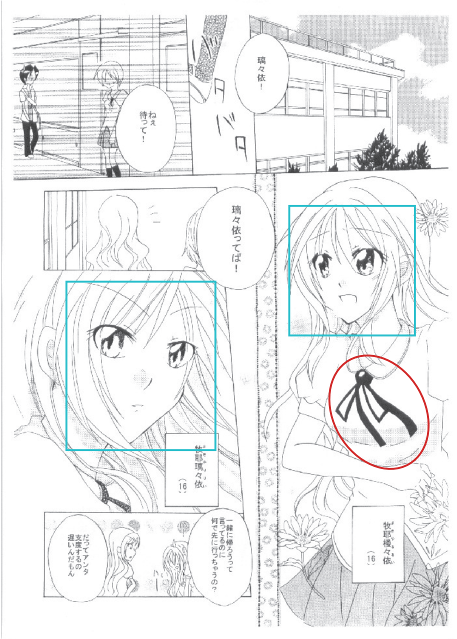
Series Retrieval by Bag-of-Features Approaches

4.1 Introduction

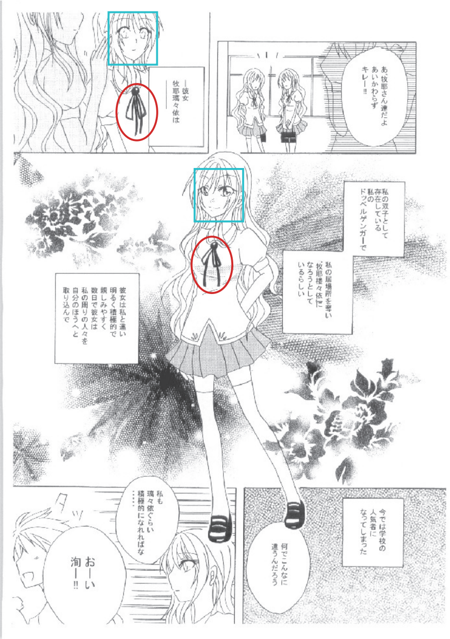
As a popular serial reading material, manga publications are released constantly, thus, the amount of manga pages that require copyright protection are enormous and increasing rapidly. Therefore, it is difficult to collect all of them to build the database for the copy detection, such as the method proposed in section 2.

From the view of the structure of serial manga publication, one series is divided into several volumes, in which hundreds of manga pages are contained, and released periodically. Since one series of manga describes a sequential narrative, some parts like the main characters are drawn in its all volumes and treated as a symbols for the series, as shown in Fig. 4.1. In this chapter, we study the discriminative features of different manga series, and propose a method to retrieve the series of unreleased publications based on these features extracted from released ones. The retrieval is based on a Bag-of-Features approach which is proposed in our previous work [88].

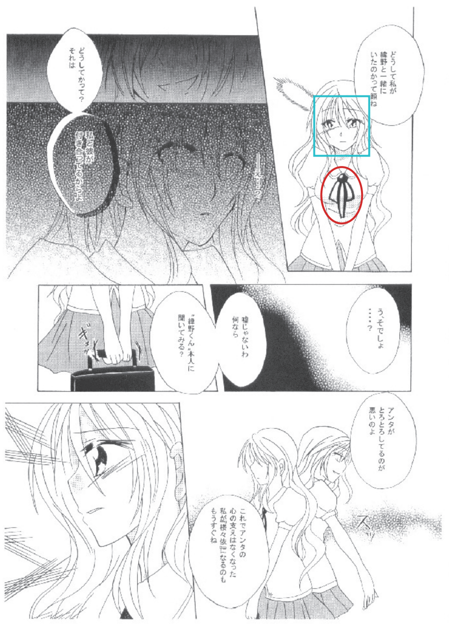
Besides manga, the method is also available for other serial publication, such as serial animated cartoons (called animations for short). Animations is a kind of artwork constructed of hand-drawn or computer-made cartoons. Most of commercial animations are sourced from popular manga and describe the narratives by a number of episodes released regularly. Compared to manga, animations can provide more vivid images through sequential frames rather than static manga pages, thus, they have a larger audience and recognition throughout the world. Traditionally, animations are distributed via television



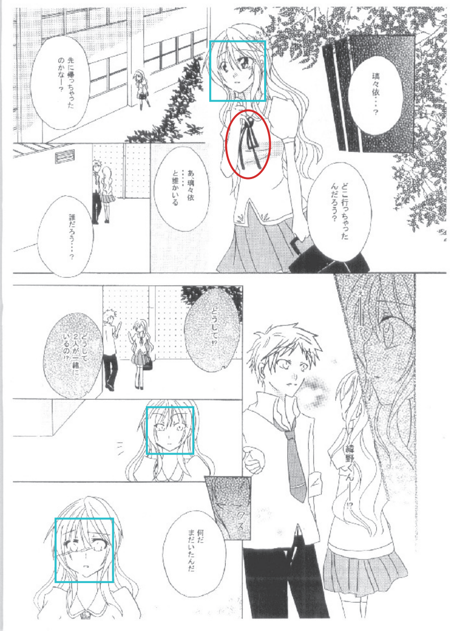
(a)



(b)



(c)



(d)

Figure 4.1: Examples of common features of one manga series. The squares mark the face regions (FRs) of the main characters. The ellipses mark generic regions (GRs) of a discriminative decoration, which is applied frequently in this series of manga.

broadcasts, directly to video, or theatrically. Recently, the development of Internet bring kinds of convenient services for animations ’ release. Whereas the copyright problem becomes more serious, which is threatening the developments of animations. To escape the detection, illegal users often divide one animation into several fragments and distribute separately. Therefore, animation fragment detection is the problem we should consider. In [89], we proposed a bag-of-features method to detect their similar fragments.

This chapter describes a bag-of-features method for series retrieval of serial publications. The purpose is to apply features of released publications to retrieve unreleased one. We take manga publication as example for analyses, but the method is also available for other serial publications. The rest parts of this chapter are arranged as follows: Section 4.2 introduces the proposed method. In section 4.3, we test the proposed method using serial manga and animations. Finally, section 4.4 is the conclusion.

4.2 Proposed method

4.2.1 Overview

For the series retrieval, we propose to apply a bag-of-features method by using the visual words based on face regions (FRs) and clusters of generic regions (GRs). The bag-of-features method is a popular method for image classification [90], video retrieval [91], object classification [92] and document retrieval [42]. It is analogous to the bag-of-words representation of text documents by an unordered set of words.

Taking serial manga publication for example, as shown in Fig. 4.2, in the proposed method, copyrighted mangas categorized by their series are collect in the database. Based on the features extracted from the database, a visual word dictionary is built. By matching the feature vectors extracted from the database with a visual word dictionary, each series of manga get a bag-of-features representation. On the other hand, a suspicious manga page is treated as a query. By the same method, the query is also represented by a bag-of-features representation. Based on the comparison of bag-of-features representations between the query and the database, the series of manga with the maximum similarity is reported as the result.

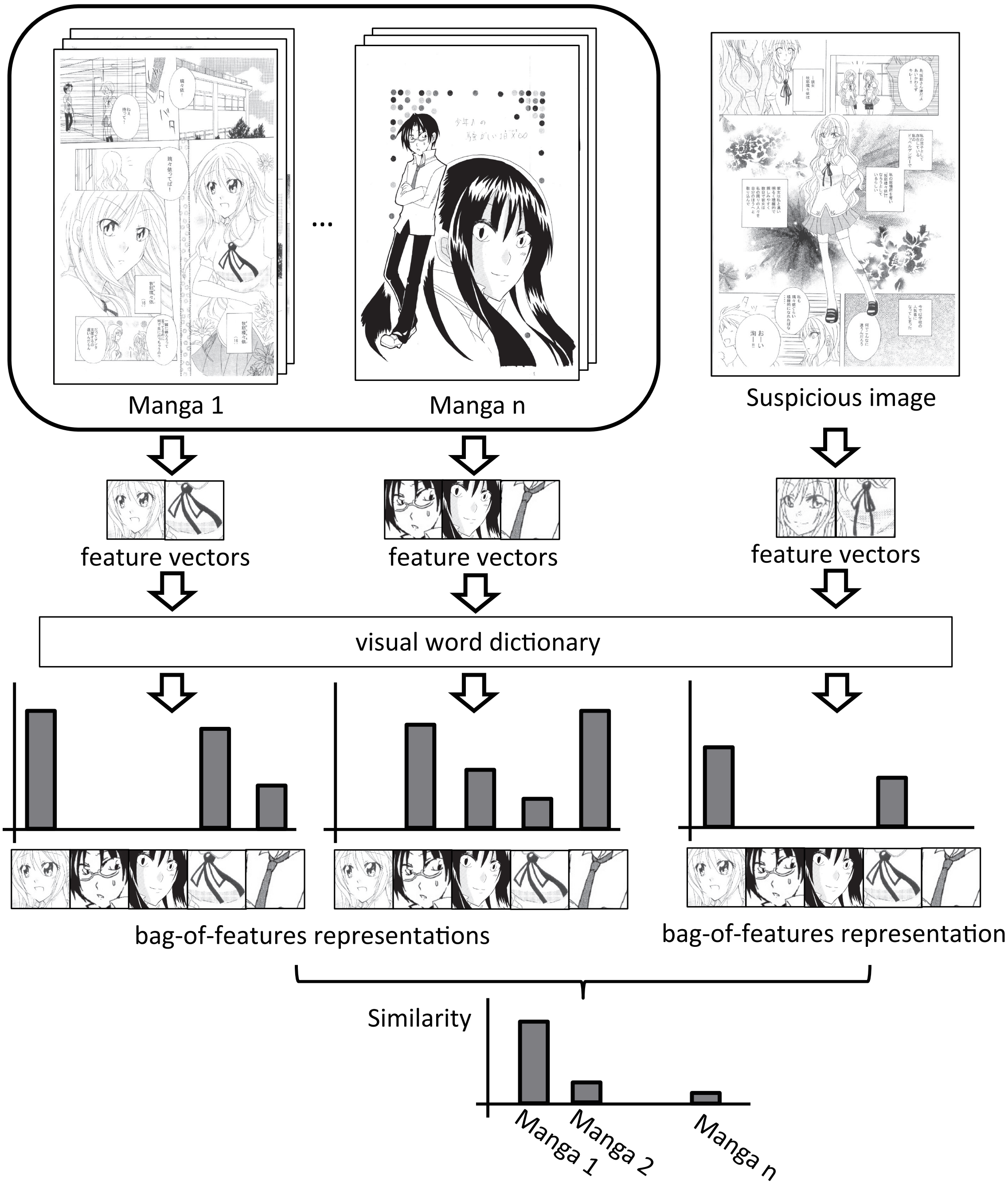
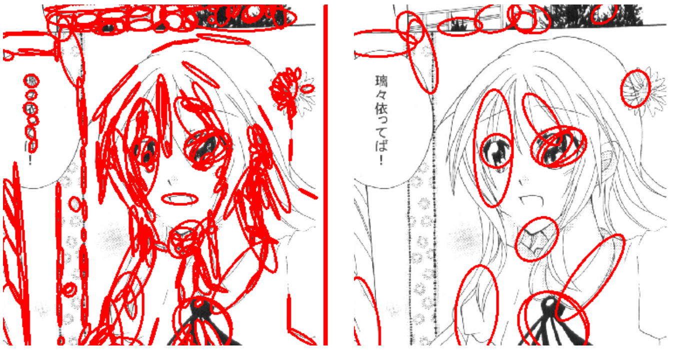


Figure 4.2: Outline of the proposed method.

4.2.2 Region detector and feature descriptor

4.2.2.1 Region Detector

In this research, we apply the features from two kinds of regions: FRs which is the regions around faces of cartoon characters and GRs which is a kind of local feature



(a) Ellipse MSERs. (b) Applied regions.

Figure 4.3: Examples of detected ellipse MSERs and applied regions.

regions.

For detecting FRs, we follow the approach applied in [68] which is described in section 3.3.2. For detecting GRs, MSER (Maximally Stable Extremal Regions) [61] is applied as the region detector, which is described in section 2.3.2. By the region detector, maximally stable regions (MSERs), in which the intensities of all pixels are greater or smaller than their boundaries, can be detected from the images. By diagonalizing the covariance matrices of MSERs, we can get some ellipse regions, as shown in Fig. 4.3(a). As shown in Fig. 4.3(b), considering the stability of GRs, only the regions above a certain size are applied. Then, we normalize the ellipse MSERs by rotating the long axis of ellipse parallel to the y axis of the image. To make the regions contain more information, both FRs and GRs are magnified as k times (in this research we set $k = 1.5$).

4.2.2.2 Feature descriptor

For the presentation of detected regions, we propose applying HOG (Histogram of Oriented Gradients) [29] as the feature descriptor.

The specific calculation of HOG is described in section 2.3.3. For each FR and GR, there is 8×8 cells and 7×7 blocks (2×2 cells per block). The gradient orientation is quantized into 6 bins. As a results, we obtain one $6 \times 2 \times 2 \times 7 \times 7 = 1,176$ dimensional HOG feature vector for each region.

4.2.2.3 Matching

For the matching of any two feature vectors F_i and F_j, we calculate the Euclidean distance D between them. In this research, we define the similarity by Euclidean distance D between feature vectors If $D < T_1$ (T_1 is a threshold), F_i is similar with F_j, otherwise not.

4.2.3 Visual words

4.2.3.1 Overview

There are many regions (FRs and GRs) detected from the publications. However, not all of them are useful for the retrieval and applied for the visual words in the dictionary. For different kinds of regions, the features are chosen for visual words as follows.

- FR

  Since the cartoon character are an essential parts of manga and animations and always treated as symbols of the series, especially the main characters, they are important for the retrieval. As the discussion in section 3.3.3.5, more possible patterns of the cartoon characters can offer a better performance of their recognition. Therefore, all the FRs are applied to make the visual word dictionary.

- GR

  From the publications, we can detect many GRs including some useless parts for the retrieval. To select discriminative GRs, we propose to detect clusters of GRs according to the dependency between GRs and series. Taking manga for example, normally, one series of manga consists of multiple volumes which contain several manga pages. Because the main parts are usually drawn repeatedly in one series, high-frequency parts should be important. However, some parts like text balloons, stripes and tones are employed frequently in many other series but not offering significant information. Therefore, we should consider the dependency between GRs and series from the two scales: the scale of a single series and the scale of different series.

The detection processing of GR clusters is shown in Fig. 4.4. First, in the processing of a single series, high-frequency GRs are detected. Then, in the processing of different

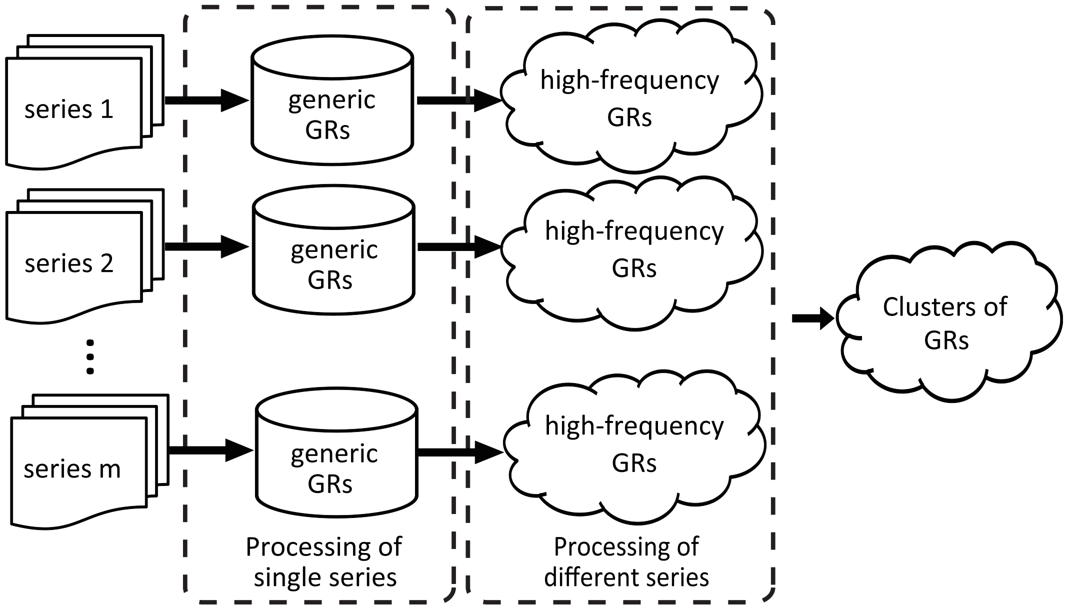


Figure 4.4: Detection processing of GR clusters

series, high-frequency GRs from different series are merged. By a further selection, high-discriminative clusters of GRs are outputted.

4.2.3.2 Processing of single series

For each series, we calculate in the feature space the neighbors of all GRs and joint similar ones into clusters by a union-find algorithm. Clusters of high-frequency GRs of series d are defined as

$$\frac{N_{c,d}}{N_d} > T_2 \tag{4.1}$$

where $N_{c,d}$ is the number of GRs in cluster c of series d, N_d is the total number of GRs from series d, T_2 is a threshold that controls the number of clusters used for next stage.

To increase the speed of distance calculation, ANN (Approximate Nearest Neighbor Searching) [50] is applied. ANN is a method to find approximate nearest neighbors by using the k-d tree. The detail is described in section 2.3.5. While searching, the feature space shrunk by the factor $1/(1+\varepsilon)$ (ε is set to be 10 in this research).

4.2.3.3 Processing of different series

In the processing of different series, first, we merge the similar high-frequency GRs from different series. Then, all the merged clusters are evaluated by considering mutual dependency between clusters and different series. We propose applying mutual information [93] to select them. Mutual information can be equivalently expressed as

$$I(X;Y) = H(X) - H(X|Y) \tag{4.2}$$

where X represents series and Y represents merged clusters, $H(X)$ is the entropy of different series, $H(X|Y)$ represents the conditional entropies of series given clusters in this method. Mutual information measures the information that X and Y share. In other words, it measures the certainty of X by knowing Y. The entropy of different series is measured by

$$H(X) = -\sum_{j=1}^{n} P(x_j) \log P(x_j) \tag{4.3}$$

where $P(x_j)$ is probability for one series j, n is the number of series. All the probabilities P applied for the calculation of entropy are based on the number of GRs. $H(X|Y_i)$ is measured by

$$H(X|Y_i) = -\sum_{j=1}^{n} P(x_j|y_i)P(y_i) \log P(x_j|y_i) \tag{4.4}$$

where $P(x_j|y_i)$ is the probability of series j given the merged cluster i, $P(y_i)$ is the probability of cluster i. By evaluating every merged clusters based on their mutual information with series, we get several clusters (Y_i) which satisfy $I(X;Y_i) > T_3$. The decision of T_3 depends on the distribution of mutual information of Y_i.

The advantages of the processing of this method are twofold: one is for considering frequency of GRs in the scales of both single series and different series, the other one is dominancy for calculations, including memory cost for distance calculations and needless recalculations of formal clusters of high-frequency GRs for additional series.

4.2.4 Retrieval

As visual words, we applied the feature vectors of FRs and the centroid of feature vectors of GRs within one cluster. By gathering the visual words, we can build a visual word dictionary. Each title of series d is represented by a bag-of-features representation

$\boldsymbol{V}_d = (R_{1,d}, R_{2,d}, ..., R_{n,d}, W_{1,d}, W_{2,d}, ..., W_{m,d})$, where n and m are the total number of visual words based on FRs and clusters of GRs, $R_{f,d}$ and $W_{c,d}$ represent weights for visual words based on FRs and clusters of GRs, respectively. $R_{f,d}$ is set to be 1 if FRs f belongs to series d, overwise 0.

For the weight $W_{c,d}$, tf-idf (term frequency-inverse document frequency) weighting [94] is employed. In our case, the GRs are terms and different series are documents. Therefore, the weight $W_{c,d}$ is calculated by

$$W_{c,d} = \frac{n_{c,d}}{n_d} \log \frac{N}{n_c} \tag{4.5}$$

where $n_{c,d}$ is the number of GRs of cluster c in series d, n_d is the total number of GRs in series d, N is the total number of series, and n_c is the number of series containing the cluster c.

The query is also represented by the bag-of-features representation $\boldsymbol{Q}$. We calculate the similarities S_d between $\boldsymbol{Q}$ and $\boldsymbol{V}_d$ as

$$S_d = \frac{\boldsymbol{Q} \cdot \boldsymbol{V}_d}{\|\boldsymbol{Q}\| \|\boldsymbol{V}_d\|} \tag{4.6}$$

The series with the maximum similarity is reported as the result.

4.3 Experimental Results

To test the effectiveness of the proposed method, we collected some series of manga and animations, a part of which are applied as released ones to build the database, and rest of which are treated as unreleased ones for the queries. In section 4.3.1, we tested the detection of GR clusters and series retrieval by using manga pages. In section 4.3.2, we utilized animations to discuss the relation of database size to performance and test effectiveness of the proposed method for animation fragments.

4.3.1 Series Retrieval of Manga

4.3.1.1 Experiment Settings

We collected 20 series of Japanese manga which contain 3 volumes for each series, and employed vol. 1 and vol. 2 as our database. It contains $6,642$ full manga pages (about $800 \times 1,200$ pixels) in total. As queries, we utilized $1,884$ manga pages from vol. 3 of

10 series of manga whose vol. 1 and vol. 2 are stored in the database. Therefore, the queries contain the similar features of their series but not exactly the same to the images stored in the database. All the pages of the manga magazines including cover pages were utilized in our experiments without any selections.

4.3.1.2 Detection of clusters of GRs

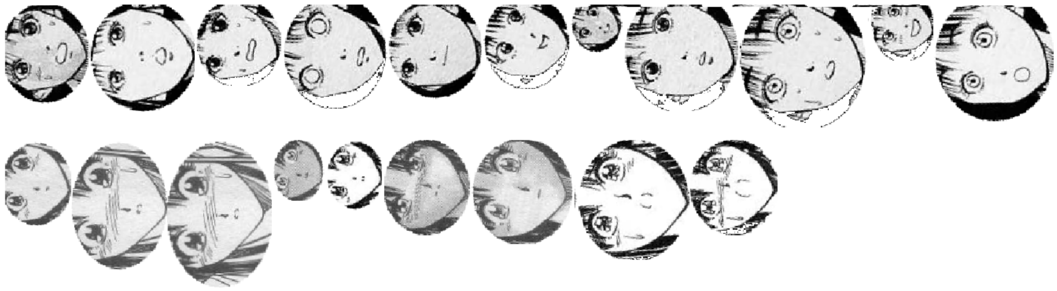
First, we did an experiment to know what kinds of clusters of GRs can be extracted by the proposed method.

By the region detector of MSER, we detected $1,376,571$ GRs. For the processing of single title of manga (T_2 is set to be 5×10^{-5}), we got $216,614$ high-frequency GRs. For the processing of different titles of manga, T_3 was set to filter half of the clusters and $11,827$ clusters were obtained. of GRs (). Examples of detected and filtered clusters of GRs are shown in Fig. 4.5 and Fig. 4.6, respectively. The GRs from different titles of manga are shown in different lines, and GRs in one line are from the same manga. From the examples, we can see:

- Effectiveness of the proposed method for clustering the similar regions. Besides scale transformations and rotations, the ROIs contain many changes in detail. Such as shown in Fig. 4.5(b), the regions are similar but not the same kanji.
- Besides faces of main characters, many other patterns have high dependency with certain manga. Such as Fig. 4.5(c) shows, they are a kind of special patterns only appeared in one manga.
- As shown in Fig. 4.6, some patterns like text balloons and numbers, which are applied frequently in many series, are less of discrimination and filtered by the proposed processing.

4.3.1.3 Series Retrieval

In this experiment, we tested the effectiveness of the proposed method for series retrieval of manga publications. As comparative methods, we employed the following 3 kinds of bag-of-features methods: (1) **GR**, which is using feature vectors extracted from all GRs as visual words, (2) **cluster of GRs**, which is only using the visual words of detected clusters of GRs, (3) **FR**, which only applied the feature vectors extracted from FRs as



(a)

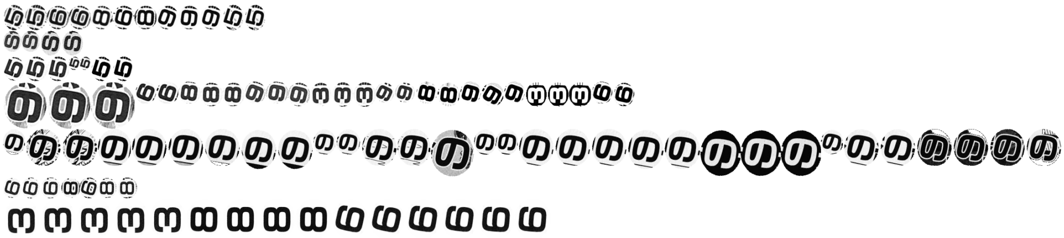
(b)

(c)

Figure 4.5: Examples of detected clusters of GRs. The lines show GRs in the scale of single series of manga. The rows show GRs in the scale of different series of manga. (There are parts of detected clusters of GRs).

visual words. Their feature descriptor and matching method were the same. The same series of manga as queries (one manga page is treated as one query) were treated as the right answers. The results were reported by recall $R = A/B$ and precision $P = A/C$, where A is the number of right answers, B is the number of queries, and C is the number of retrieved queries. Since the matching of visual words is closely related to threshold T_1, we did the retrieval experiments by applying different threshold T_1 and got the results as shown in Fig. 4.7.

With the increase of T_1, because we can obtain more FRs and GRs for both right and erroneous matchings, the precisions decreased with the increase of recall. For a small T_1, only regions with few differences were matched. Since the method based on clusters of GRs applied the centroid of feature vectors extracted from GRs, the visual words are different from original feature vectors. Therefore, for a small T_1 the method based on clusters of GRs performed worse than the other three methods. Above a certain T_1, the



(a)

(b)

Figure 4.6: Examples of filtered clusters of GRs. The lines show GRs in the scale of single title of manga. The rows show GRs in the scale of different titles of mangas. (There are parts of detected clusters of GRs).

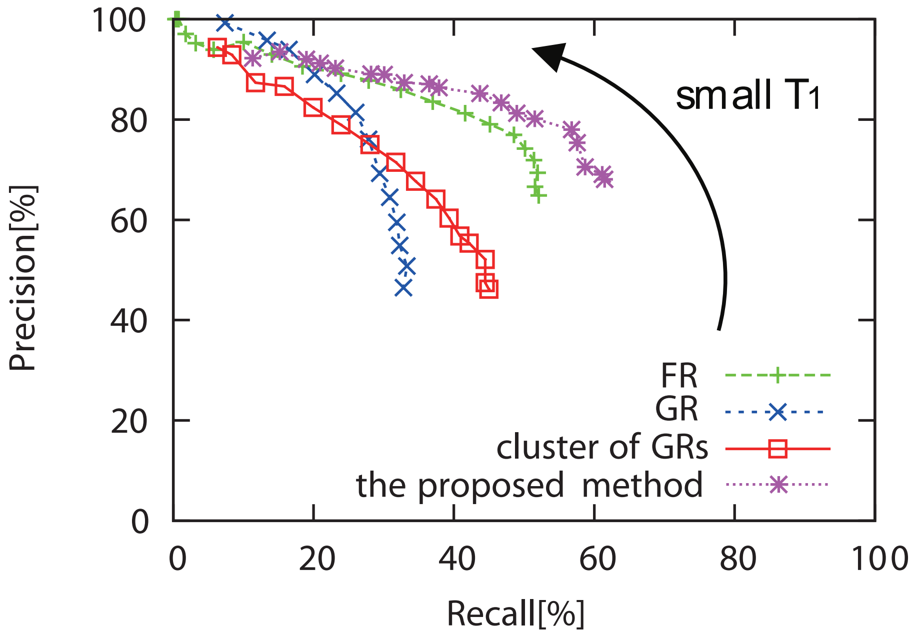


Figure 4.7: Precision and recall for series retrieval of manga.

Table 4.1: Max F-scores of different methods.

GRs	20.9
cluster of GRs	23.8
FRs	30.0
the proposed method	31.4

matchings began to be untrusted and more erroneous visual words came out from queries, which decreased both precision and recall.

The max F-scores of different methods are shown in Table ??. From the results, we can see

- The clusters of GRs detected by the proposed method are more discriminative than GRs without clustering. Therefore, comparing the method using all GRs, the method using clusters of GRs performed better.

- FRs are one kind of important regions for manga. Since face patterns are highly discriminative and included frequently in manga, the method only using FRs showed high performance for the retrieval. Although there are some face patterns in clusters of GRs, the performance of clusters of GRs did not outperform the method only using FRs, because of the limited number and less discriminative power of face patterns in clusters of GRs.

- Besides FRs, other GRs also offer us a clue for series retrieval. Therefore, the proposed method performed best in the 4 methods, even better than the method only using FRs for recall up to 10%.

The main reasons for failures are: (1) some manga pages, such as cover pages, do not contain discriminative GRs, (2) limited database images for detecting the clusters of GRs. Because of the complexities of manga, more training data are required.

4.3.2 Series Retrieval of Animations

In this section, we applied animations, which are another important serial publications, to test the proposed method. Compared to manga, animations apply sequence of images within a duration time, thus, queries offer us more information for the retrieval. After the description of experiment setting in section 4.3.2.1, we discuss the relation between database size and performance in section 4.3.2.2, and test the retrieval of animation fragments with different duration time.

4.3.2.1 Experiment Setting

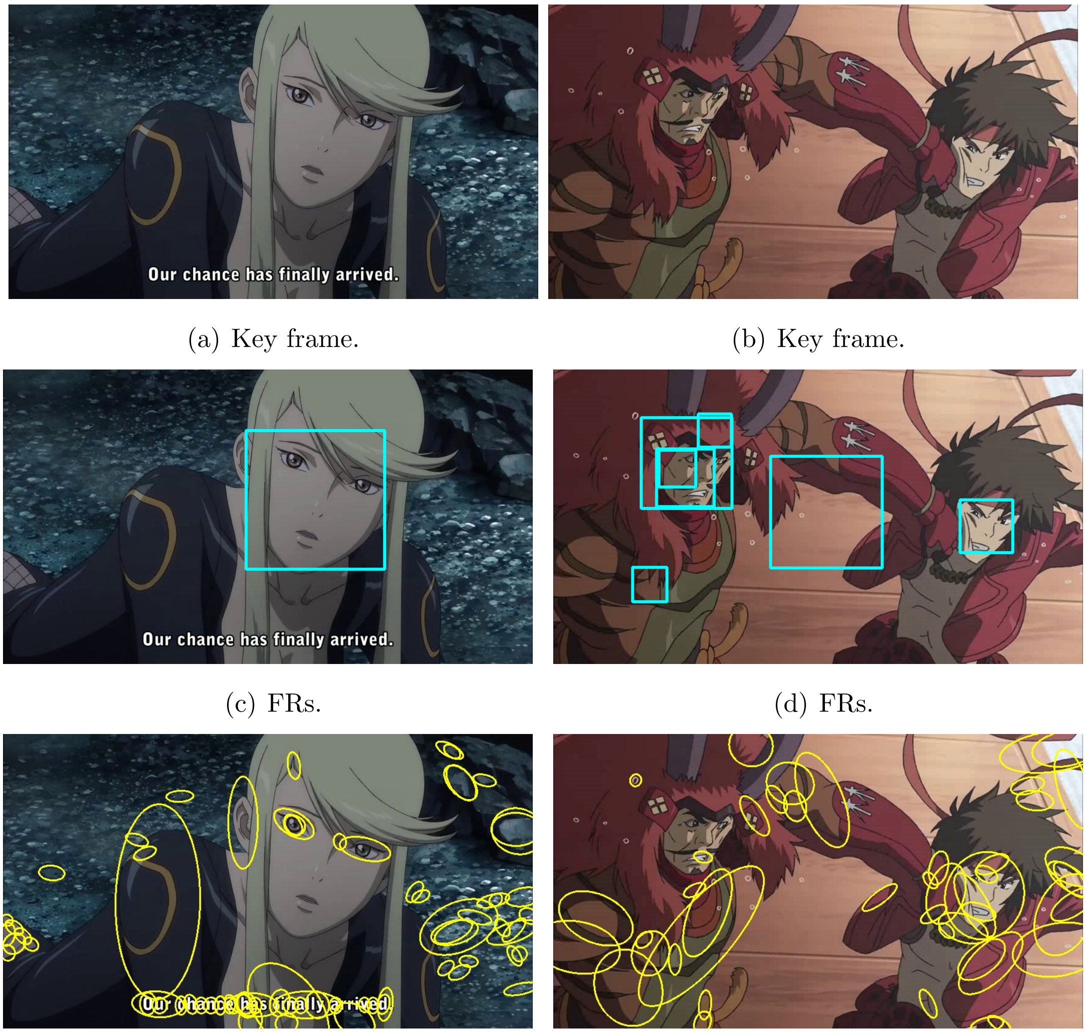
The processing is the same to the method applied in manga, but the feature detection is based on key frames instead of manga pages. The key frames are extracted in a certain sampling rate, which is set to one frame per 2 seconds in this experiment. For detecting FR, we trained the detector with $4,000$ positive samples (face regions of cartoon character) and $10,000$ negative samples (non-face regions) using animation frames (the detail of the detection is described in section 3.3.2). The setting of GRs was the same. The example of key frames and the applied regions are shown in Fig. 4.8.

Totally, 12 series of animations were collected. We kept their resolutions as their releases, and all the resolutions are higher than 640×480. For each series, there are five episodes and one episode lasts about 25 minutes. From these animations, the first three episodes are applied to build the databases. The rests (Eps. 4 and Eps. 5) are divided into fragments of a certain duration (t) evenly as queries. Therefore, the queries are similar to the ones with the same series in the databases, but not exactly the same. The same series as queries were treated as the right answers, thus there is one and only one right answer for each query.

The matching of visual words is closely related to the threshold T. For a small T, since only regions with few differences were matched, we can get more reliable matching for retrieval and achieve a high precision. With the increase of T, both right and erroneous matchings are increased, and thereby the precision decreased with the increase of recall. In the experiments, T was changed and chosen for the best performance. All experiments were done with a computer of INTEL i7-870 2.93GHz CPU and 8 GB RAM.

4.3.2.2 Experiment 1

First, we tested the effect of the database size. We build three databases of different sizes (DB1 contains Eps. 1, DB2 contains Eps. 1 and 2, DB3 contains Eps. 1, 2 and 3). Based on the three databases, we built three visual word dictionaries and did retrievals. As a benchmark, queries ($t = 70$ seconds) are applied. The interpolated average precisions are shown in Fig. 4.9. From the results, we can conclude that more discriminative visual words obtained from a larger data set, by which the proposed method achieved a better



(a) Key frame. (b) Key frame.

(c) FRs. (d) FRs.

(e) GRs (f) GRs.

Figure 4.8: Examples of detected FRs and GRs from key frames. (They are cited from animation "Basara".)

performance for the series retrieval.

4.3.2.3 Experiment 2

Then, we chose the DB3 to test retrieval for fragments of different duration time. For the queries, t was set to 2, 20, 70, 140, 200, 400, 700 seconds. There is only 1 key frame for each query ($t = 2$ sec.), since key frames are extracted in every 2 seconds. The results are shown in Fig. 4.10. Since queries of longer duration time can offer us more frames also clues for the retrieval, both recalls and precisions were increasing with longer t. By using

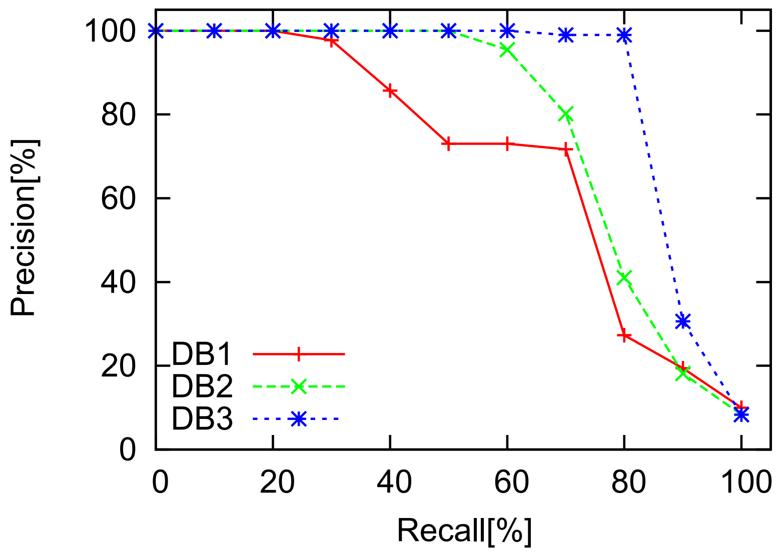


Figure 4.9: Interpolated average precision-recall curves for different data sets.

a smaller T_1, we can achieve high precision with low recall matching of regions. Based on the abundant regions offered by animation fragments, we achieved high recall with high precision retrieval. For the queries without enough discriminative regions inside, the recall is relative low. The detection time increases for longer durations, since more features are applied. For queries ($t = 140$ ms), the average processing time (excluding time for feature extraction) is $2,682$ ms.

The main reasons for failures are described as follows. (1) Since not all frames contain a discriminative information, some animation fragment with few matched regions for retrieval. (2) Animations always apply color feature to classify objects. In this research, we only use shape features to represent the regions. Therefore, there are some erroneous matchings. As shown in Fig. 4.11, the two characters are similar in shape but with different hair color.

4.3.3 Discussion

From the experiments, we can see that

- Face region is an important part and discriminative features of manga and animations.
- The proposed method can detect discriminative clusters from GRs.

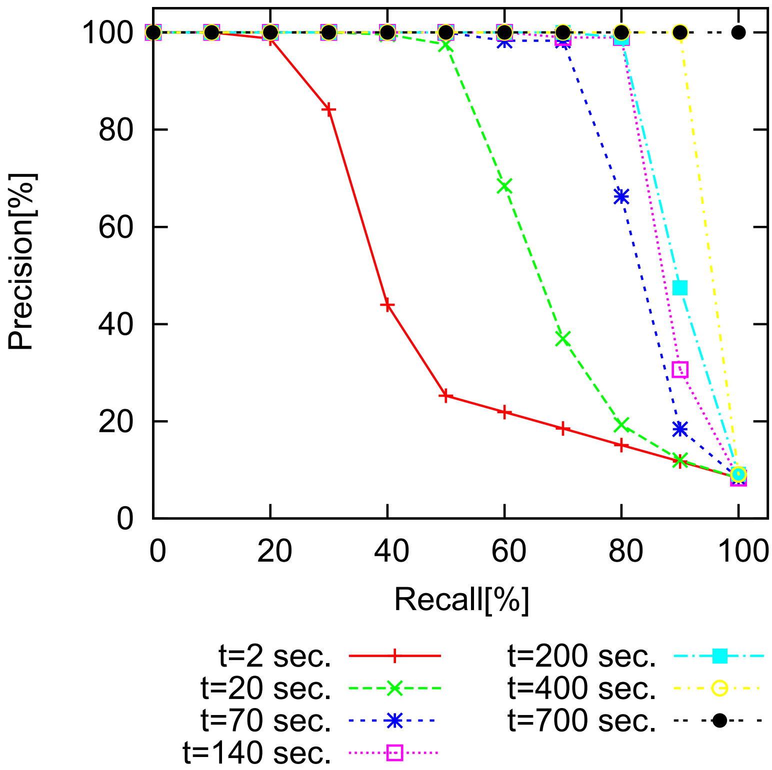
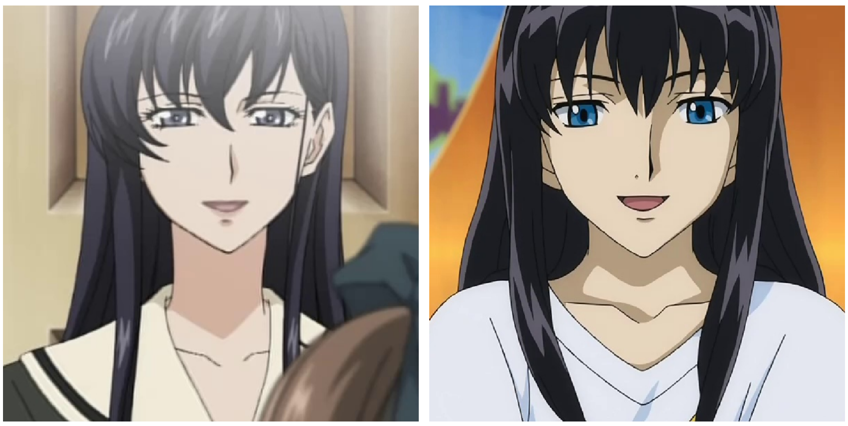


Figure 4.10: Interpolated average precision-recall curves for fragments of different durations.



(a) (b)

Figure 4.11: Example of similar characters. The color of their eyes and hair are different. (Fig. 4.11(a) is from Eps. 3 of "Marimite". Fig. 4.11(b) is from Eps. 4 of "Gundam Series".)

- One kind of serial publication has some significant features, by which we can retrieve the publications belong to the same series.

- Larger dataset of each series for building the visual word dictionary can lead to a better performance.

- For animation, fragments with longer duration time can offer more clue for the retrieval.

4.4 Conclusion

In this chapter, we proposed a bag-of-features method for series retrieval of serial publication. For the visual words, features of two kinds of regions (FRs and GRs) are applied. In addition, we proposed a method to detect discriminative clusters of GRs to enhance the effectiveness of the proposed method. From the experimental results, we proved the effectiveness of the proposed method for serial manga publications. In addition, we applied the animation for the experiments, and discussed the relation of training dataset to performance. Depending on the plenty information offered by the queries of animations, we achieved a impressive results by using high-precision matchings. Through the experiments, we reveal that (1) besides the main characters, there are many other parts can offer clues of series, (2) based on the features of release publication, the unreleased ones can be retrieved.

Our future work includes

- increasing the recall and precision of manga page retrieval,

- employing other regions for the visual words,

- use color features for the region representations of animations,

- increasing the data sets for training,

- considering methods to increase the retrieval.

Chapter 5

Conclusion

This thesis describes our studies on copy detection of line drawings for their copyright protection. We focus on manga, one of the most important line drawing publications, and proposed applying Content-Based Copy Detection (CBCD) approach to detect their copies. Because of the simple compositions and abstract expressions applied in manga, there are some unusual types of copies like hand-drawn copies and similar copies, which issue challenges to the typical CBCD methods. For the problems, we proposed the stable region detection and discriminative feature representation methods. In addition, database construction and fast detection methods are studied for the practical applications.

In chapter 2, we focused on the detection of exact partial copies, which copy a specific part of certain images, and proposed using local feature matching. For the local feature regions, regions surrounded by lines are applied, because of their robustness for the variations caused by hand drawing. HOG (Histogram of Orientation Gradients), which is an effective feature descriptor for handwriting recognition, is employed for the representations of the local feature regions. In addition, we studied the reliable matchings and proposed a robust matching method. As the results, not only printed but also hand-drawn partial copies can be detected from unknown backgrounds with their corresponding original parts reported. Moreover, for practical applications, we also studied the reduction of database and feature vector indexing methods, and made an improvement of database construction, by which we achieved a better performance with a smaller database. In the experiments, a database of $10,009$ manga pages were applied to test the proposed method. Even the partial copies only occupy 10% of the while images, we achieved 99% precision at 95% recall for printed copy detection, and 42% precision at 85% recall for hand-drawn copy detection within 1.5 seconds per image.

In chapter 3, we described our studies on the detection of similar cartoon characters.

Since the similar copies of cartoon characters are created based on the different images of a specific character not a specific part of images, the method described in chapter 2 is not available for their detection. Considering the faces are a basic part of cartoon characters, we propose applying face regions as Regions of Interest (ROIs) for their detection and recognition. In the proposed method, first, cartoon characters are located in the suspicious manga pages by an object detection method. Based on the detected ROIs from copyrighted images, we can build a database. Then, the cartoon characters detected from suspicious manga pages are recognized by matching the similar ones in the database. However, since the faces of cartoon characters contain few identifiable features with many variations, such as facial expressions, viewpoints and occlusions, it is difficult to recognize them only depending on features extracted from the face regions. For this problem, we proposed Concentric Multi-Region (CMR) model to explore the significant features from the parts around face regions, and trained a new feature named Concentric Multi-Region Histograms of Oriented Gradients (CMR-HOG) by HOG feature representation and the AdaBoost algorithm. To test the effectiveness of the proposed method for cartoon character recognition, 17 labeled cartoon characters are applied in the experiment. Compared to other high-performance face and object recognition methods based on face regions, the proposed method shows a better performance for the recognition of cartoon characters. Furthermore, we make a discussion by combining the detection with recognition. Based on the experimental results, the proposed method can achieve a 82.2% precision with 75.9% recall for detecting similar copies of cartoon characters within 165 ms per character from 20 series of manga.

In chapter 4, we introduced the research on series retrieval of serial manga. As described in chapters 2 and 3, the CBCD methods require databases of originals. However, since the popular manga or animations are serial publications whose amount are enormous and increasing rapidly, it is difficult to collect the whole series to build the database. Since there are some discriminative parts appear through the whole series, we propose to retrieve the series of unreleased publications (will be released in the future) based on the features from released ones. For this purpose, we study the features of serial publications, and proposed a bag-of-features (BOF) approach to retrieve their series. Features proposed in chapters 2 and 3 are applied. For this purpose, BOF model is applied. We also proposed a method to detect discriminative features for the visual word dictionary. In the experiments, we applied 20 series of manga publications and achieved the retrieval of manga pages in vol. 3 based on the visual words from vol. 1 and vol. 2. Moreover, 12 series of animations was applied to discuss the relation of training dataset to the performance of the retrieval. By using high-precision matching, we achieve 98% precision with 80% for the animation fragments of 140 seconds in 2.7 second.

As stated above, we proposed CBCD methods to detect various copies of manga. For the problem of illegal copy, although judgment of professionals is an essential process, the proposed methods can reduce the manual workload while assisting the professionals with original information as evidence. For copy detection systems used in practice, recall means the power to find copies, and precision means how much they can reduce manual work. Based on the experimental results, given 50% as an acceptable precision, which means two results need to be checked manually for one real copy, above 75% copies can be detected by the proposed method. From this point, we can see the contribution of the proposed method for the copyright protection of manga.

In addition, for both building the databases and detection of copies, manga pages are used directly without selection, cropping or any supervision in our system. This is a practical feature for actual use.

Following tasks are included in the future work.

- Increase the database of copyrighted manga pages.

  Also, we need to increase the scalability of the proposed method, since there are enormous amount of manga requiring copyright protection,

- Speeding up the detection.

  In practical application, the proposed method is applied to detect all suspicious images, which is far more than the queries applied in our experiments. The detection speed is an important effectiveness factor for the copy detection systems.

Acknowledgement

I would like to thank sincerely my supervisor Professor Koichi Kise for his guidance and encouragement.

I wish to thank Professor YoshiOka and Professor Miyamoto for valuable advises and helpful suggestions.

I would like to acknowledge Assistant Professor Masakazu Iwamura for his encouragement and kind assistance.

Thanks are due to all members of Intelligent Media Processing Group of Osaka Prefecture University for their assistance.

I have to save my final thought for my family. They have always stood behind me and have provided me with immeasurable encouragement and support. They have been there through all the trials and tribulations for my graduate education. They give me the strength and determination to forge ahead through the most difficult times.

Bibliography

[1] "2007 publication index annual report." The all Japan Magazine and Book publishers' and editors Association, the Research Institute for Publications, 2002.

[2] T. Matsui, "The diffusion of foreign cultural products: The case analysis of japanese comics (manga) market in the us," *Princeton University, Woodrow Wilson School of Public and International Affairs, Center for Arts and Cultural Policy Studies, working paper*, vol. 37, Jan. 2009.

[3] J. Fishbein, "Europe's manga mania," *BusinessWeek*, Nov. 2007.

[4] "ebook business 2008 report." Impress R&D, 2009.

[5] http://www.kidsandadvertising.co.uk/cartoon-characters-kids-advertising.html.

[6] http://www.animevice.com/characters/.

[7] C.-Y. Lin, M. Wu, S. Member, J. A. Bloom, I. J. Cox, S. Member, M. L. Miller, and Y. M. Lui, "Rotation, scale, and translation resilient watermarking for images," *IEEE Trans. Image Process.*, vol. 10, pp. 767–782, May 2001.

[8] P. Bas, J.-M. Chassery, and B. M. Macq, "Geometrically invariant watermarking using feature points," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 1014–1028, 2002.

[9] A. Nikolaidis and I. Pitas, "Region-based image watermarking," *IEEE Trans. Image Process.*, vol. 10, pp. 1726–1740, Nov. 2001.

[10] S. Pereira and T. Pun, "Robust template matching for affine resistant image watermarks," *IEEE Trans. Image Process.*, vol. 9, pp. 1123–1129, June 2000.

[11] D. Dori, "Orthogonal zig-zag: an algorithm for vectorizing engineering drawings compared with hough transform," *Adv. Eng. Softw.*, vol. 28, pp. 11–24, Jan. 1997.

[12] J. Song, F. Su, C. L. Tai, and S. Cai, "An object-oriented progressive-simplification based vectorization system for engineering drawings: model, algorithm, and performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 1048–1060, Aug. 2002.

[13] S. H. Joseph and T. P. Pridmore, "Knowledge-directed interpretation of mechanical engineering drawings," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 9, pp. 928–940, 1992.

[14] T. Lu, C.-L. Tai, H. Yang, and S. Cai, "A novel knowledge-based system for interpreting complex engineering drawings: theory, representation, and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1444–1457, 2009.

[15] J. Lladós, E. Martí, and J. López-Krahe, "A hough-based method for hatched pattern detection in maps and diagrams," in *Proceedings of the 5th International Conference on Document Analysis and Recognition*, pp. 479–482, 1999.

[16] R. Kasturi, S. T. Bow, S. Member, S. Member, W. El-masri, J. Shah, J. R. Gattiker, Umesh, and B. Mokate, "A system for interpretation of line drawings," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, pp. 978–992, 1990.

[17] Z. Li, M. Schulte-Austum, and M. Neschen, "Fast logo detection and recognition in document images," in *Proceedings of the 20th International Conference on Pattern Recognition*, pp. 2716–2719, 2010.

[18] S.-A. Berrani, L. Amsaleg, and P. Gros, "Robust content-based image searches for copyright protection," in *Proceedings of the 1st ACM international workshop on Multimedia databases*, pp. 70–77, 2003.

[19] D.-Q. Zhang and S.-F. Chang, "Detecting image near-duplicate by stochastic attributed relational graph matching with learning," in *Proceedings of the 12th ACM International Conference on Multimedia*, pp. 877–884, 2004.

[20] Y. Ke, R. Sukthankar, L. Huston, Y. Ke, and R. Sukthankar, "Efficient near-duplicate detection and sub-image retrieval," in *Proceedings of 12th ACM Conference of Multimedia*, pp. 869–876, 2004.

[21] J.-H. Hsiao, C.-S. Chen, L.-F. Chien, and M.-S. Chen, "A new approach to image copy detection based on extended feature sets," *IEEE Trans. Image Process.*, vol. 16, pp. 2069–2079, Aug. 2007.

[22] W.-L. Zhao and C.-W. Ngo, "Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection," *Trans. Image Process.*, vol. 18, pp. 412–423, Feb. 2009.

[23] M. Douze, H. Jegou, and C. Schmid, "An image-based approach to video copy detection with spatio-temporal post-filtering," *IEEE Trans. Multimedia*, vol. 12, pp. 257–266, June 2010.

[24] A. Joly, C. Frélicot, and O. Buisson, "Robust content-based video copy identification in a large reference database," in *Proceedings of the 2nd International Conference on Image and Video Retrieval*, pp. 414–424, 2003.

[25] A. Joly, C. Frélicot, and O. Buisson, "Content-based copy retrieval using distortion-based probabilistic similarity search," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 293–306, 2007.

[26] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.

[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, Nov. 2004.

[28] Y. Ke and R. Sukthankar, "Pca-sift: a more distinctive representation for local image descriptors," in *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pp. 506–513, 2004.

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

[30] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the 2nd European Conference on Computational Learning Theory*, EuroCOLT '95, pp. 23–37, 1995.

[31] W. Sun and K. Kise, "A method for copyright protection of line drawings," pp. 167–168, Aug. 2009.

[32] W. Sun and K. Kise, "Detecting printed and handwritten partial copies of line drawings embedded in complex backgrounds," in *Proceedings of the 10th International Conference on Document Analysis and Recognition*, pp. 341–345, 2009.

[33] W. Sun and K. Kise, "Speeding up the detection of line drawings using a hash table," vol. 2, pp. 896–900, Nov. 2009.

[34] W. Sun and K. Kise, "Partial copy detection for copyright protection of line drawings (in japanese)," *IEICE trans. Information and Systems*, vol. J93-D, pp. 909–919, June 2010.

[35] C.-Y. Chiu, C.-S. Chen, and L.-F. Chien, "A framework for handling spatiotemporal variations in video copy detection," vol. 18, pp. 412–417, Mar. 2008.

[36] J. Han and K. Ma, "Fuzzy color histogram and its use in color image retrieval," *IEEE Trans. Image Process.*, vol. 11, pp. 944–952, Aug. 2002.

[37] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Computer Vision*, vol. 7, pp. 11–32, 1991.

[38] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, pp. 837–842, Aug. 1996.

[39] M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance," *IEEE Trans. Image Process.*, vol. 11, pp. 146–158, 2002.

[40] S. Berretti, A. D. Bimbo, and P. Pala, "Retrieval by shape similarity with perceptual distance and effective indexing," *IEEE trans. MultiMedia*, vol. 2, no. 4, pp. 225–239, 2000.

[41] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 509–522, April 2002.

[42] M. R. nol and J. Lladós, "Logo spotting by a bag-of-words approach for document categorization," in *Proceedings of the 10th International Conference on Document Analysis and Recognition*, pp. 111–115, 2009.

[43] C. Kim and B. Vasudev, "Spatiotemporal sequence matching for efficient video copy detection," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 15, pp. 127–132, Jan. 2005.

[44] S. Smith, "A new class of corner finder," in *Proceedings of the 3rd International Conference of British Machine Vision*, pp. 139–148, 1992.

[45] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 530–535, May 1997.

[46] S. Eickeler and S. Mller, "Content-based video indexing of tv broadcast news using hidden markov models," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2997–3000, 1999.

[47] M. Heikkila, M. Pietikainen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognition*, vol. 42, pp. 425–436, Mar. 2009.

[48] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *Int. J. Comput. Vision*, vol. 60, pp. 63–86, Oct. 2004.

[49] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, June 1981.

[50] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *J. ACM*, vol. 45, pp. 891–923, Nov. 1998.

[51] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the 25th International Conference on Very Large Data Bases*, pp. 518–529, 1999.

[52] R. Salakhutdinov and G. Hinton, "Semantic hashing," *Int. J. Approx. Reasoning*, vol. 50, pp. 969–978, July 2009.

[53] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proceedings of Neural Information Processing Systems*, pp. 1753–1760, 2008.

[54] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proceedings of VISAPP International Conference on Computer Vision Theory and Applications*, pp. 331–340, 2009.

[55] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[56] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[57] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter sensitive hashing," in *Proceedings of IEEE International Conference on Computer VIsion*, vol. 2, pp. 750–757, 2003.

[58] W. Sun and K. Kise, "Detection of exact and similar partial copies for copyright protection of manga (reviewing)," *Int. J. Document Analysis and Recognition*, 2012.

[59] C. Harris and M. Stephens, "A combined corner and edge detection," in *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, 1988.

[60] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *J. Comput. Vis. Image Underst.*, vol. 110, June 2008.

[61] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proceedings of the 18th British Machine Vision Conference*, pp. 384–393, 2002.

[62] R. Sedgewick, *Algorithms.* Addison-Wesley, 2nd ed., 1988.

[63] Q. Zhu, Q. Zhu, S. Avidan, S. Avidan, M. chen Yeh, M. chen Yeh, K. ting Cheng, and K. ting Cheng, "Fast human detection using a cascade of histograms of oriented gradients," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1491–1498, 2006.

[64] I. Jolliffe, *Principal Component Analysis.* Springer Verlag, 1986.

[65] K. Kise, K. Noguchi, and M. Iwamura, "Simple representation and approximate search of feature vectors for large-scale object recognition," in *Proceedings of the 18th British Machine Vision Conference*, vol. 1, pp. 182–191, 2007.

[66] W. Sun and K. Kise, "Copyright protection of line drawings: copyrighted part detection using cascade classifiers," *Proceedings of the 2nd China-Japan-Korea Joint Workshop on Pattern Recognition*, pp. 9–10, Nov. 2010.

[67] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, pp. 137–154, May 2004.

[68] W. Sun and K. Kise, "Similar partial copy detection of line drawings using a cascade classifier and feature matching," in *Proceedings of the 4th international conference on Computational forensics*, pp. 126–137, 2011.

[69] W. Sun and K. Kise, "Similar partial copy recognition for line drawings using concentric multi-region histograms of oriented gradients," *Proceedings of the 12th IAPR Conference on Machine Vision Applications*, pp. 71–74, May 2011.

[70] W. Sun and K. Kise, "Cartoon character recognition using concentric multi-region histograms of oriented gradients (reviewing)," *J. Institute of Electrical Engineers of Japan*, 2012.

[71] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 34–58, 2002.

[72] G. Yang and T. S. Huang, "Human face detection in a complex background," *Pattern Recognition*, vol. 27, no. 1, pp. 53–63, 1994.

[73] H. P. Graf, E. Cosatto, D. Gibbon, M. Kocheisen, and E. Petajan, "Multi-modal system for locating heads and faces," in *In Proc. 2nd Int. Conf. Automatic Face and Gesture Recognition*, pp. 88–93, 1996.

[74] I. Craw, H. Ellis, and J. Lishman, "Automatic extraction of face-features," *Pattern Recognition Letters*, vol. 5, no. 2, pp. 183–187, 1987.

[75] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, pp. 23–38, Jan. 1998.

[76] B. Heisele, T. Serre, and T. Poggio, "A Component-based Framework for Face Detection and Identification," *Int. J. Comput. Vision*, vol. 74, pp. 167–181, Aug. 2007.

[77] F. Fleuret and D. Geman, "Coarse-to-fine face detection," *Int. J. Comput. Vision*, vol. 41, pp. 85–107, Jan. 2001.

[78] H. Schneiderman and T. Kanade, "Object detection using the statistics of parts," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 151–177, 2004.

[79] S. Z. Li and Z. Zhang, "Floatboost learning and statistical face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, pp. 1112–1123, Sept. 2004.

[80] M. Zhou and H. Wei, "Face verification using gaborwavelets and adaboost," in *Proceeding of the 18th International Conference on Pattern Recognition*, vol. 1, pp. 404–407, 2006.

[81] O. Déniz, G. Bueno, J. Salido, and F. De la Torre, "Face recognition using histograms of oriented gradients," *Pattern Recognition Letter*, vol. 32, pp. 1598–1603, Sept. 2011.

[82] C. Liu, "Gabor-based kernel pca with fractional power polynomial models for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, pp. 572–581, 2004.

[83] Y. Qin and G. Sun, "Handwritten character recognition using multi-resolution histograms of oriented gradients," in *Proceedings of IEEE International Conference on Communication Technology*, pp. 715–717, 2008.

[84] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the ACM International Conference on Image and Video Retrieval*, 2007.

[85] J. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, pp. 1148–1161, Nov. 1993.

[86] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Wurtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Trans. Computers*, vol. 42, pp. 300–311, Mar. 1993.

[87] Z. Li, J.-i. Imai, and M. Kaneko, "Facial-component-based bag of words and phog descriptor for facial expression recognition," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 1353–1358, 2009.

[88] W. Sun and K. Kise, "Similar manga retrieval using visual vocabulary based on regions of interest," *Proceedings of the 11th International Conference on Document Analysis and Recognition*, pp. 1075–1079, Oct. 2011.

[89] W. Sun, K. Kise, and Y. Champeil, "Similar fragment retrieval of animations by a bag-of-features approach," *Proceedings of the 10th IAPR International Workshop on Document Analysis Systems*, pp. 140–144, Mar. 2012.

[90] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *Proceedings of the 9th European Conference on Computer Vision*, vol. 4, pp. 490–503, 2006.

[91] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings of 9th IEEE International Conference on Computer Vision*, vol. 2, pp. 1470–1477, Oct. 2003.

[92] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proceeding of ECCV Workshop on Statistical Learning in Computer Vision*, pp. 1–22, 2004.

[93] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing.* MIT Press, 1999.

[94] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc., 1999.