Similar Partial Copy Detection of Line Drawings Using a Cascade Classifier and Feature Matching

Weihan Sun, Koichi Kise

Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka, 599-8531 Japan
sunweihan@m.cs.osakafu-u.ac.jp
kise@cs.osakafu-u.ac.jp

Abstract. Copyright protection of image publications is an important task of forensics. In this paper, we focus on line drawings, which are represented by lines in monochrome. Since partial copies and similar copies are always applied in plagiarisms of line drawings, we propose combining the technique of object detection and image retrieval to detect similar partial copies from suspicious images: first, detecting regions of interest (ROIs) by a cascade classifier; then, locate the corresponding source parts from copyrighted images using a feature matching method. The experimental results have proved the effectiveness of proposed method for detecting similar partial copies from complex backgrounds.

Keywords: Line drawing, Copyright protection, Object detection, Image retrieval, Similar copy, Partial copy

1 Introduction

The development of computer techniques offers us methods to store images in digital mode and distribute quickly through the Internet. In contrast of conveniences, it also causes copyright problems of images. Such as comics, graphs and logos, line drawings are an important part of image publications. Line drawings are a type of images that consist of distinct straight and curved lines in monochrome or few colors placed against plain backgrounds. Because of simplicities of line drawings, it is easy to create similar drawings. Practically, illegal users usually copy the important part and apply them as a part of their own drawings, which are called partial copies. Therefore, for protecting the copyright of line drawings, we should consider the detection of similar copies and partial copies, which bring more challenges to their copyright protection.

In our previous research, we proposed applying the technique of image retrieval to detect partial copies of line drawings [1]: copyrighted line drawings are collected in a database, and suspicious images are treated as queries. By applying MSER (Maximally Stable Extremal Regions) [2] as a region detector and HOG (Histograms of Oriented Gradients) [3] as a feature detector, we achieved detecting both printed and handwritten partial copies from complex backgrounds. However, because of enormous volume of line drawing publications which require copyright protection, the problems of database overload and low detection speed prevent the method to be utilized in practice.

Considering not the whole image requires copyright protection, the key to solve these problems is only to store important parts in our database. In addition, based on the important parts in the database, the similar copies should also be able to be detected. However, which part is the important part? This is a quite controversy problem. Here, let us focus on one of the most important line drawing publications: comics, in which faces of characters might be the parts which requires copyright protection.

To detect such important parts (regions of interest (ROIs)), we propose applying an object detection technique. There are many researches about object detection, such as the detection of face, pedestrian, vehicle and so on [3–5]. Viola et al. [6] proposed a framework to train a cascade classifier, by which rigid objects can be detected with fast detection speed and low false positive rate. However, since terrific transformation in faces of comic characters, the effectiveness for detecting comic faces has not been proved. In this paper, we apply Viola-Jone framework to line drawings and prove it is available for detecting the faces of comic characters. Furthermore, we built a database using detected parts and achieved the detection of similar parts by applying a method of feature matching.

The rest parts of this paper is arranged as: Section 2. describes our proposed method. Experiments and results is shown in Section 3. Finally, Section 4. is conclusions and future works.

2 Proposed method

2.1 Overview

As shown in Fig. 1, the processing of the proposed method is divided into two parts: ROI detection and similar part detection. In the part of ROI detection, we propose applying the method of object detection to detect possible faces of comic characters. Then, in the part of similar part detection, the detected parts from copyrighted images are collected to build a database. The parts extracted from suspicious images are treated as queries. By matching the features of queries and the database, the part which is similar with each query is reported.

2.2 ROI detection

To detect faces of comic characters, we propose applying Viola-Jones detection framework. In this framework, positive samples (images contain the object) and negative samples (non-object images) are collected. Then, features are extracted from these samples and marked with their response (1:object, 0:non-object). After that, the most characteristic features are selected. Based on a threshold, a decision tree can be built for each feature, which is called a classifier. By arranging such classifiers in a cascade structure, we can get a cascade classifier.

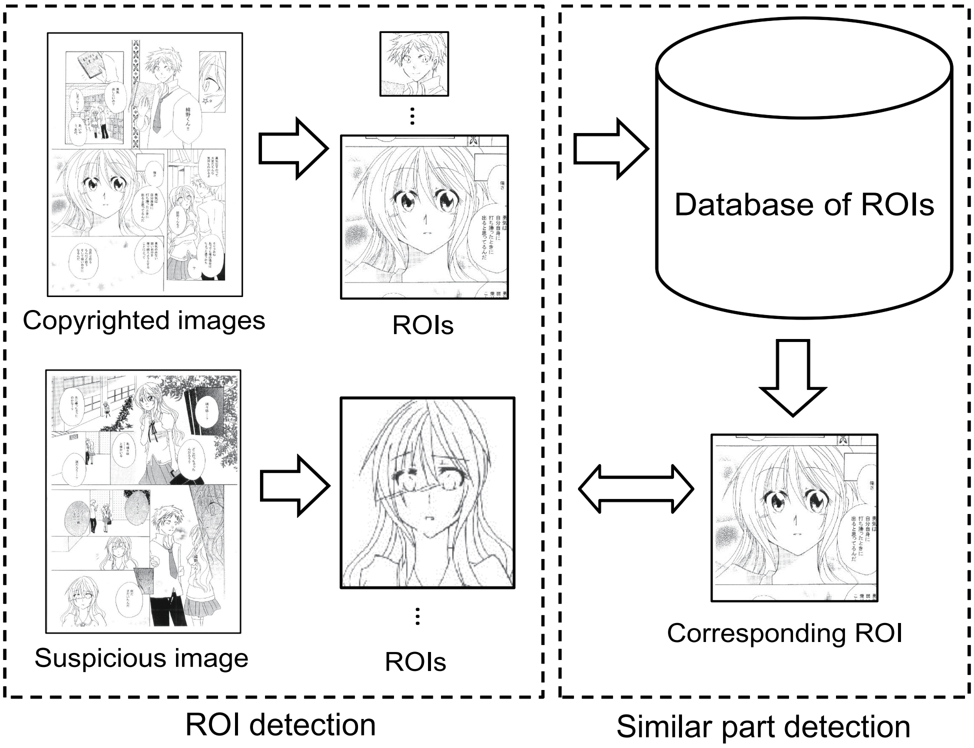


Fig. 1. Processing of the proposed method.

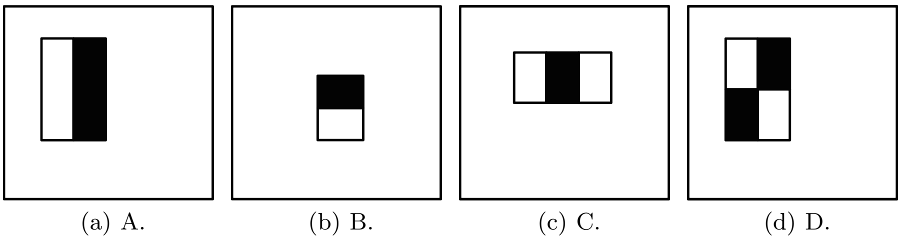


Fig. 2. Example of using Haar-like features.

Feature extraction Viola et al. applied Haar-like features in their method. More specifically, they utilized four kinds of features. As show in Fig. 2, 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 (detector). We will have $162,336$ features for a detector with the size of 24×24 pixels.

To speed up the calculation of Haar-like features, Viola et al. proposed applying integral image. The integral image at location x, y contains the sum of

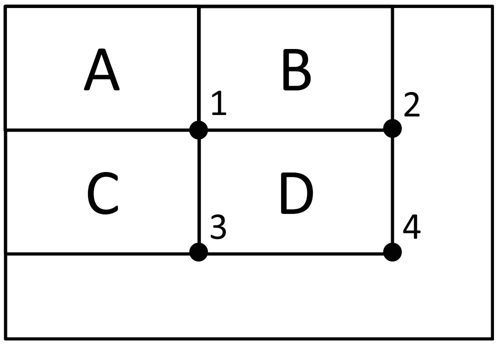


Fig. 3. Calculation using integral image. The value at location 1 (ii_1) is the sum of the pixels in rectangle A, ii_2 is A+B, ii_3 is A+C, ii_4 is A+B+C+D. Therefore, the sum pixels in D can be computed as $ii_4 + ii_1 - ii_2 - ii_3$.

the pixels above and to the left of x, y, inclusive:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

where $ii(x, y)$ is the integral image and $i(x, y)$ is the original image. As shown in Fig. 3, we can get the sum of pixels for any region by 3 times' calculations. Therefore, the Haar-like features can be calculated in a constant time.

Also other features can be applied to the Viola-Jones framework. A more precise description of objects' characteristics can lead to a higher detection rate. In addition, since the construction of cascade classifiers demands many samples, the speed of feature calculation has an important effect on training time.

Feature selection Since the Haar-like feature set is over-complete, Adaboost [7] is utilized for choosing the features. AdaBoost is a machine learning method to combine weak classifiers into a strong classifier by an iterative algorithm. The specific algorithm is shown in Fig. 4. In each loop, a weak classifier with the lowest error rate is generated, and the samples which failed to be classified will be assigned a larger weight to increase the precision of following classifiers. Finally, weak classifiers (with weights which depend on accuracy) are added to be a strong classifier. In Viola-Jones framework, weak classifiers are decision trees built by different features in one detector. According to a certain threshold, weak classifiers predict the results (1: positive, 0 negative).

Construction of cascade classifiers Furthermore, the framework imports a cascade structure to increase the detection performance while reducing computation time. As shown in Fig. 5, only the sub-windows predicted true can go to next layers.

The Adaboost algorithm is applied to build such a cascade automatically. The specific algorithm is shown in Fig. 6. By reducing the threshold of classifiers,

1. Given example images (x_1,y_1),...,(x_n, y_n) where $y_i = 0$ for negative samples and $y_i = 1$ for positive samples.
2. Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for each sample, where m and l are the number of negatives and positives respectively.
3. For t = 1,...,T:
   (a) Normalize the weights, $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
   (b) Select the best weak classifier with respect to the weighted error

$$\epsilon_t = min_{f,p,\theta} \sum_i w_i |h(x_i, f, p, \theta) - y_i|$$

   (c) Define $h_t(x) = h(x, f_t, p_t, \theta_t)$ where f_t, pt_t, and θ_t are the minimizers of ϵ_t.
   (d) Update the weights:

$$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$$

   where $e_i = 0$ if sample x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
4. The final strong classifier is:

$$C(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t x \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

Fig. 4. Boosting algorithm.

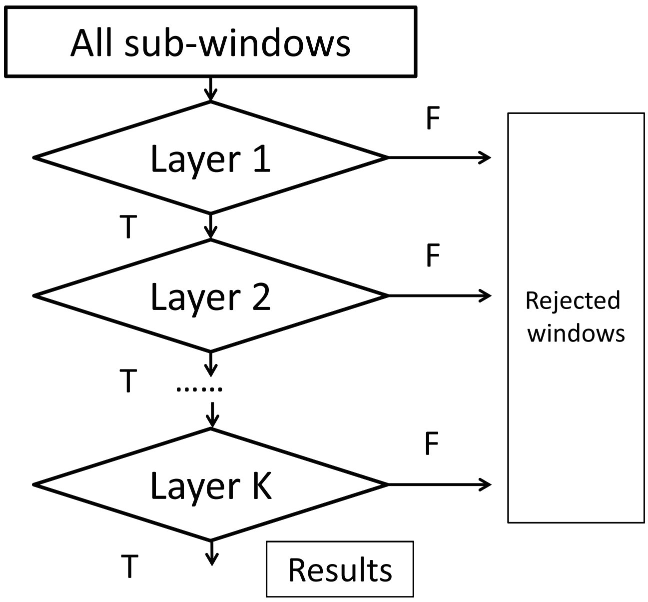


Fig. 5. Detection using cascade classifiers.

- User selects values for f, the maximum acceptable false positive rate per layer and d, the minimum acceptable detection rate per layer.
- User Selects target over all false positive rate, F_{target}.
- P = set of positive samples
- N = set of negative samples
- $F_0 = 1, D_0 = 1$
- $i = 0$
- while $F_i > F_{target}$
  - $i \leftarrow i + 1$
  - $n_i = 0, F_i = F_{i-1}$
  - while $F_i > f \times F_{i-1}$
    - $n_i \leftarrow n_i + 1$
    - Use P and N to train a classifier with n_i features using AdaBoost
    - Evaluate current cascaded classifier on validation set to determine F_i and D_i.
    - Decrease threshold for the ith classifier until the current cascaded classifier has a detection rate of at least $d \times D_{i-1}$ (this also affects F_i)
  - $N \leftarrow \emptyset$
  - If $F_i > F_{target}$ then evaluate the current cascaded detector on the set of non-face images and put any false detections into the set N

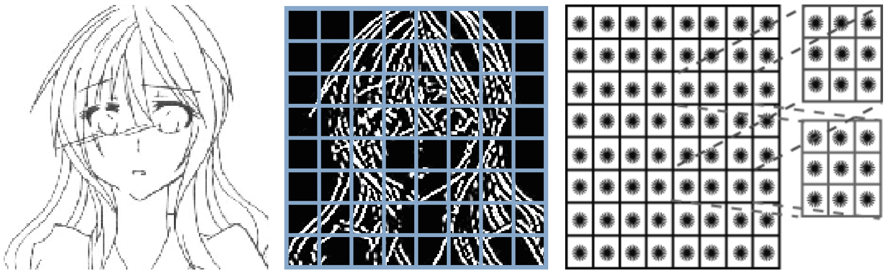
Fig. 6. Cascade detector training algorithm.

it is easy to achieve a high detection rate with a high false positive rate. 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 d_i, f_i represent detection rate and false positive rate for each layer respectively, K is the number of cascade layers. For example, for a 10 layers cascade classifier, if d_i is set to be 0.995 and f_i is set to be 0.3 for each layer. For the final cascade classifier, the detection rate will be about 0.95(since $0.995^{10} \approx 0.95$) and the false positive rate will be about 6×10^{-6} (since $0.3^{10} \approx 6 \times 10^{-6}$), by which can lead satisfied detections.

Detection of ROIs We apply sliding window technique to detect sub-windows in different scale. The detecting sub-windows are normalized into the same size of detector. After detecting all sub-windows of the whole image, we apply the union-find algorithm to group detected sub-windows. To make detected parts contain more information, the sub-windows are enlarged as k times large as the detected parts to be ROIs. In following parts of this paper, we set $k = 4$ by considering the balance between information and noisy in such regions.



(a) ROI. (b) Cells. (c) Blocks.

Fig. 7. Region instruction of HOG features.

2.3 Similar part detection

In the processing of similar part detection, we propose applying the HOG feature descriptor to describe the ROIs, and matching the pair of parts whose features are near each other.

Feature extraction We apply HOG to describe the detected ROIs, and extract one HOG feature from each ROI. As shown in Fig. 7(b), first calculate the gradient magnitude and the direction at each pixel, and divide each ROI into 8×8 cells evenly. Then, as shown in Fig. 7(c), the gradient directions are quantized into 9 bins. Thus we get a vector of 9 dimensions for each cell by calculating the gradient direction histogram based on the gradient strength. Next, combine the cells into overlapped blocks as 3×3 cells per block. The vector for each block is composed of vectors of cells, and the vector of each ROI consists of all normalized vectors of blocks. Therefore we extract a vector of $9 \times 3 \times 3 \times 6 \times 6 = 2916$ dimensions.

Matching Matching of queries and parts in database is based on the distance between their feature vectors. The pair of image parts which are nearest with each other is reported as result.

To speed up the searching of feature vector closed to each other, we apply ANN (Approximate Nearest Neighbor Search) [8]. ANN is a method to find the approximate nearest neighbor by using the k-d tree. To increase the matching speed, ANN searches the feature space shrunk by the factor $1/(1+\epsilon)$. Considering the detetion speed and detection accuracy, we set ϵ to 1 empirically.

3 Experiments

3.1 ROI detection

First, we did an experiment to test the effectiveness of comic face detection.

To train the cascade classifier, we collect $3,000$ frontal faces of comic characters from 20 kinds of comics. As shown in Fig. 8, most of the faces are cropped

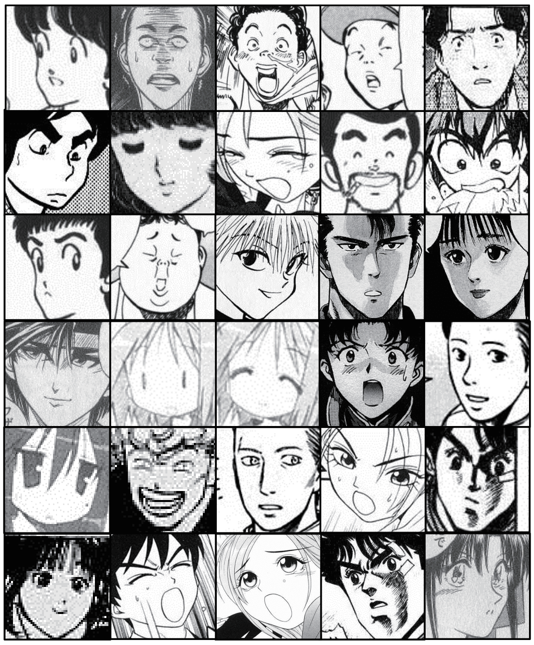


Fig. 8. Examples of positive samples used for training. They are from Rurouni Kenshin, Neon Genesis Evangelion, Hoshin Engi, H2, Hunter × Hunter, JoJo's Bizarre Adventure, Lucky Star, Master Keaton, Maison Ikkoku, Miyuki, Monster, Planetes, Rosario + Vampire, Rough and Slam Dunk.

from just above the eyebrows to chin, and normalized to 24×24 pixels as our positive samples. For the negative samples, we prepare backgrounds from comic pages without faces.

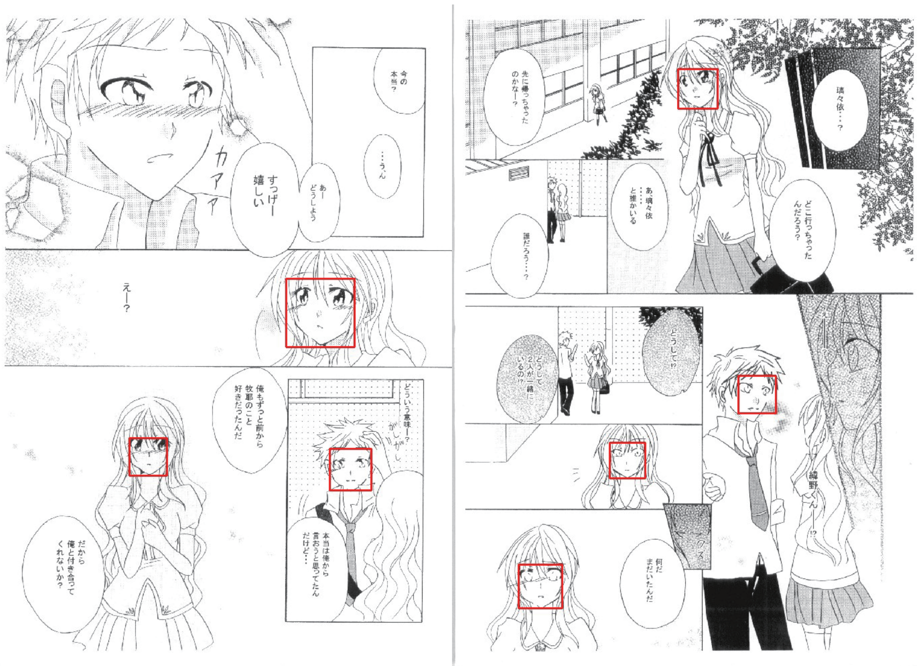
As the validate set, we chose 201 comic pages $(700 \times 1,000$ pixels) which are not utilized in the training part. There are 705 faces in the validate set.

Considering the training time, we set the cascade as 20 layers with 0.995 detection rate and 0.5 false positive rate for each layer.

The experimental results are shown in Table 1. By the cascade classifier (trained by $3,000$ positive samples and $8,000$ negative samples), we got 658 parts, in which contain 584 true faces. Examples of detection are shown in Fig. 9. The average detection time is 492 ms per image (CPU: 2.5 GHz, RAM: 4 GB).

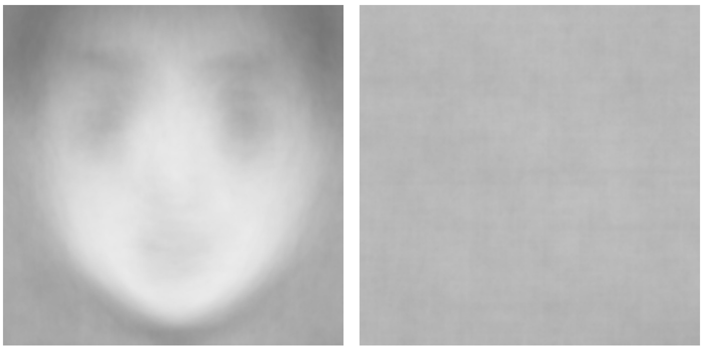
Table 1. Face detection result.

Number of detected parts	658
Number of detected faces	584
Precision	88.7%
Recall	82.8%



(a) (b)

Fig. 9. Examples of comic face detection.



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

Fig. 10. Mean images of training samples.

From the results, we can see the Viola-Jones framework is also applicable for detecting comic character's faces. Although faces of comic characters contain more variations comparing with real faces of human beings, there are still some discriminative features. As shown in Fig. 10, we can recognize the shape of face from the mean image of positive samples.

3.2 Similar part detection

Next, we tested the similar part detection of the proposed method.

As the copyrighted images, we utilized $7,682$ comic pages ($700 \times 1,000$ pixels, from volume 1 and volume 2 of 21 kinds of comics). The cascade classifier trained in the first experiment was applied detecting the possible face parts from images. From the copyrighted images, we detected $19,509$ possible face parts , and built a database with these parts.

As the query images, we chose 201 comic pages (from volume 3 of each kind comic), which contain 705 faces in total. Of course, for the same comic, the main character should appear in different volume but always with various kinds of different poses and face expressions, which are treated as similar copies in our experiments.

The experimental results are shown in Table 2. The correct matching is defined as a matched pair that belong to the same character of the same comic. All the detected parts (including right faces and non-face) are treated as queries. This means the errors of ROI detection are also included in this experimental results. Since the evaluation is strict, the precision and recall are about 50%. The examples of correct matching are shown in Fig. 11. We can see the proposed method can detect similar parts with certain range of transformations.

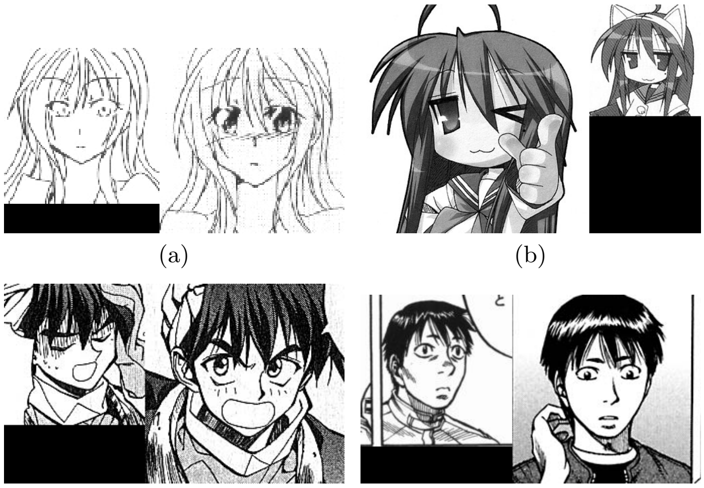
Table 2. Results of similar part detection.

Number of detected parts	658
Number of correct matching	348
Precision	52.9%
Recall	49.4%

The examples of failure are shown in Fig. 12. There are several reasons for the failures:

- Since we just utilize two volumes of each kind of comic, there are some faces of queries, which are not included in our database. For example, as shown in Fig. 12(a), some faces are with strange expressions, and some faces not from main characters as Fig. 12(b) shown.
- There are some similar features between different comic characters. Such as Fig. 12(c) and Fig. 12(d). the characters are very similar in different comic drawn by the same author.
- Some errors are caused in ROI detection. As shown in Fig. 12(e) and Fig. 12(f), the parts extracted from query are not faces.

The detection time is 68 ms per part (CPU: 2.5 GHz, RAM: 4 GB).



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

Fig. 11. Examples of correct detected similar pairs. (left : parts from queries, right : parts from database). They are from Lucky Star, Hoshin Engi and Planetes.

4 Conclusion and Future works

In this paper, we propose a method to detect illegal copies of line drawings. By applying a cascade classifier and feature matching, we have achieved the detection of similar partial copies from complex backgrounds to some extent.

There are some remaining work for us in the future including:

- detection of other kinds of ROIs beside comic faces,
- increase the detection rate of ROIs,
- describe the difference in detail of ROIs,
- enlarge the database of copyrighted images.

Acknowledgement

This research was supported in part by the Grant-in-Aid for Scientific Research (B)(22300062) from Japan Society for the Promotion of Science (JSPS).

References

1. W. Sun and K. Kise, “Detecting Printed and Handwritten Partial Copies of Line Drawings Embedded in Complex Backgrounds”, International Conference on Document Analysis and Recognition, pp.341–345, 2009.
2. J. Matas, O. Chum, M. Urban and T. Pajdla, “Robust Wide Baseline Stereo from Maximally Stable Extremal Regions”, British Machine Vision Conference, pp.384–393, 2002.

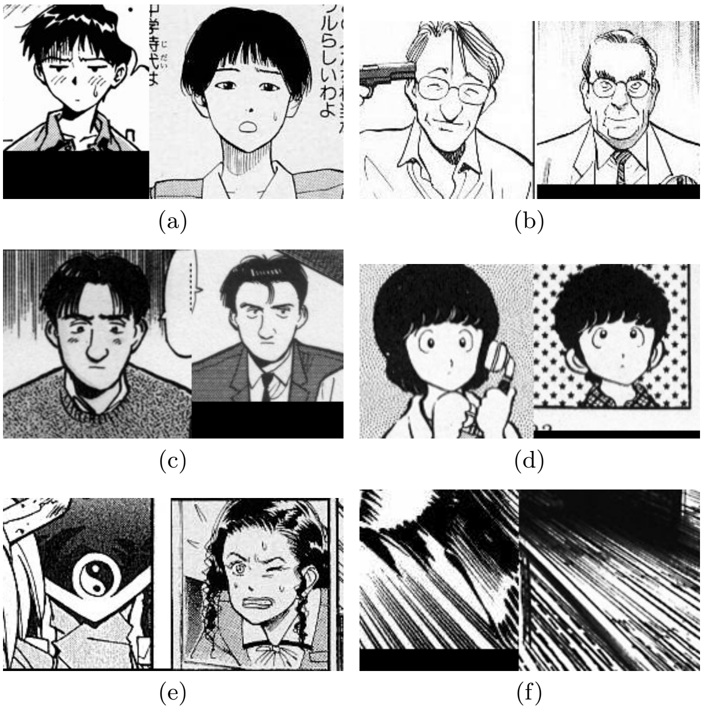


Fig. 12. Examples of failure in similar part detection. (left : parts from queries, right : parts from database) They are from Neon Genesis Evangelion, Hoshin Engi, Master Keaton, Miyuki, Monster, Slam Dunk and Rough.

3. N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", IEEE Conference on Computer Vision and Pattern Recognition, vol.1, pp.88–893, 2005.
4. Q. Zhu, S. Avidan, M. Yeh and K. Cheng, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients", IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, pp, 1491–1498, 2006.
5. H. Schneiderman and T. Kanade, "A statistical Method for 3D Object Detection Applied to Faces and Cars", IEEE Conference on Computer Vision and Pattern Recognition, 2000.
6. P. Viola, M. Jones. "Robust Real-Time Face Detection", International Journal of Computer Vision, 57(2), pp.137–154, 2004.
7. Y. Freund and R. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting", Journal of Computer and System Sciences, 55(1), pp.119–139, 1997.
8. S. Arya, D. Mount, R. Silverman and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching", Journal of the ACM, 45, 6, pp.891–923, 1998.