# KUBOI — Element

# <3 figures, 3 tables>

# Element Detection in Japanese Comic Book Panels

Kuboi, T. and Khosmood, F.

Comic books are a unique and increasingly popular form of entertainment combining visual and textual elements of communication. This work pertains to making Japanese comic books more accessible. Specifically, this paper explains how we detect elements such as speech bubbles and character eyes present in Japanese comic book panels. Some possible applications of the work presented in this paper are automatic detection of text and its transformation into audio or into other languages. Automatic detection of elements can also allow reasoning and analysis at a deeper semantic level than what’s possible today.1 We present our process for speech bubble detection in this paper.

### 

### Background

A page in a comic book consists of multiple panels, in which the scenes of the comic are depicted. Each panel has multiple elements, including text and graphics. Text in comic books is typically enclosed in speech bubbles. We use elements as a term broader than ‘object’.

Prior works by others have focused on separating text and graphics in documents (Wong et al., 1982), but not on comic book panels. Our work is important for the digital humanities community because we cover an art form that the community has not covered before.

### 

### Approach

We use a heuristic-based expert system, implemented by the authors, and machine learning for detecting elements present in images. Figure 1 shows a block diagram of our speech bubble detection pipeline. The images are segmented by the image processing system, which segments an image into regions and extracts candidate regions that are likely to contain elements of interest, which are text. Components of connected black pixels are profiled into text and non-text. Then the statistics of the candidate regions are computed, and the statistics are used to formulate the heuristics. Some of the statistics used include whether the region is enclosed by black pixels and the percentage of the area of the image the region occupies. The regions are classified whether they are elements of interest by the expert system based on the heuristics.

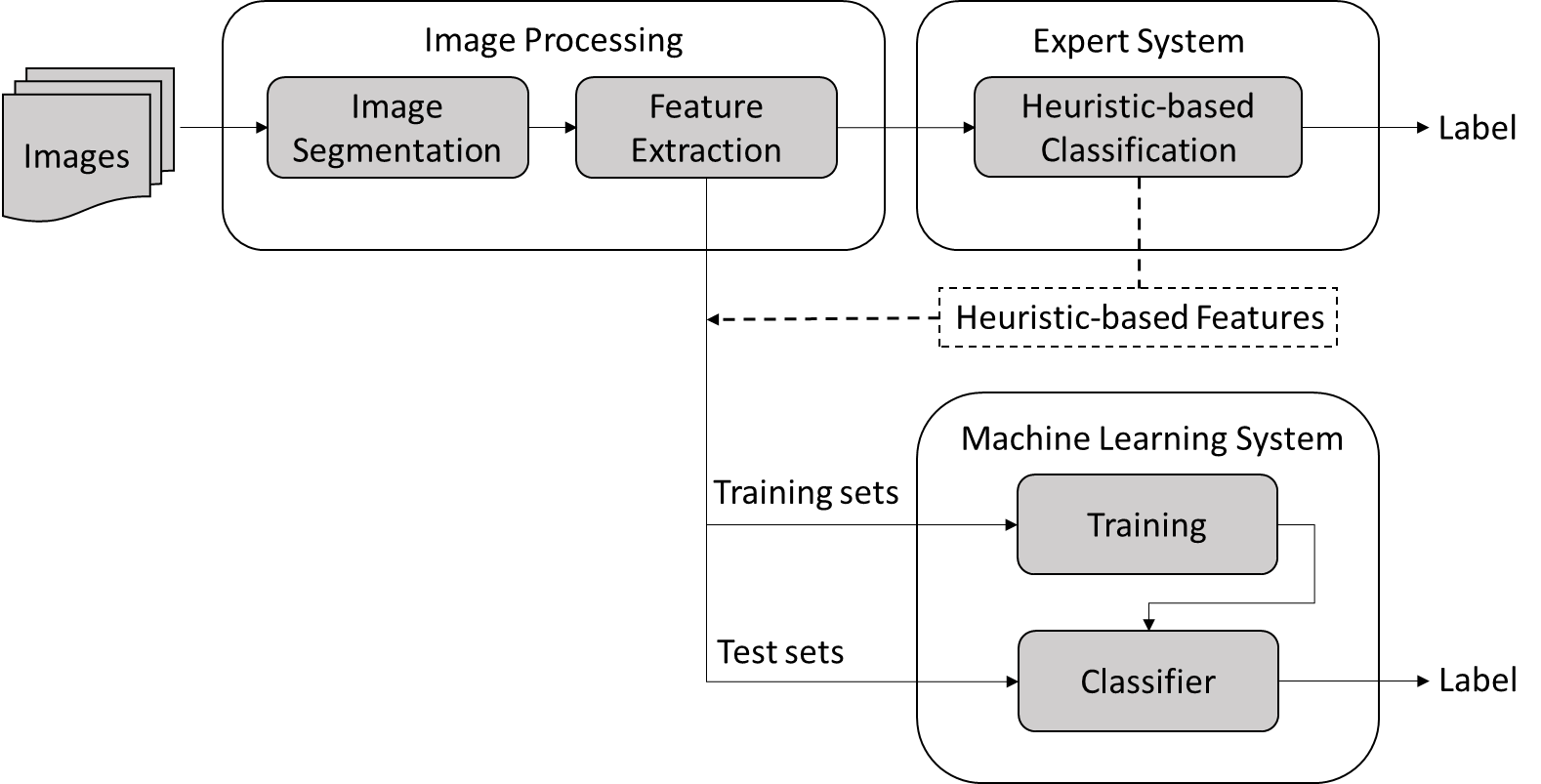


Figure 1. The Speech Bubble Detection pipeline.

We use Naive Bayes and Maximum Entropy from NLTK (NLTK Project, 2014), and support vector machine (SVM) from WEKA (Hall et al., 2009) as our learning algorithms to detect speech bubbles because of their performance and ease of use. We compute statistics of the speech bubble candidate regions in the images. For example, we compute the number of black pixels as well as the ratio between black pixels and white pixels in the regions. We also compute the size of the region in terms of the percentage of the area of the image the region occupies, and the horizontal and vertical range of the region. We use the information as the features on which the machine learning algorithms are trained. Some of the features used are heuristic-based features inspired by the expert system. Those features are binary features, such as whether the size of the region falls within a certain range, as well as whether the region is enclosed by the edges of black pixels. We allow gaps between edges that are smaller than 3 pixels wide. We also compute the vertical and horizontal distribution of the black pixels as histograms of black pixel counts and use them as features. In total, 58 features are used.

We use supervised learning and semi-supervised learning methods. In supervised learning, each candidate region is labeled as either positive or negative by humans. A positive sample means the region is a speech bubble. In semi-supervised learning, not all the training data used is hand labeled. Some of the training data is produced by the system itself. In our approach, unseen samples labeled by the trained machine learning system are added to the training data.

### 

### Speech Bubble Detection

Two hundred forty-three panels from *Doraemon*, a popular Japanese comic book—which amount to the total of 804 samples—are used to evaluate the expert system and the machine learning system. Each panel contains zero, one, or multiple speech bubbles.

Table 1 shows the performance of the expert system on the panels. The expert system achieves 95% accuracy with 91.3% precision, 97.7% recall, and the F-score of 0.944. Since the possible applications of our work require no speech bubbles lost in the process, high recall is more important than high precision.

Table 1. The performance of the expert system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Precision | Recall | F-score |
| Expert | 0.95 | 0.913 | 0.977 | 0.944 |

The same 243 panels used in the evaluation of the expert system are used for evaluation of the machine learning system. The 804 samples are divided in a 2-to-1 ratio into 539 training sets and 266 test sets. The Max Entropy is trained multiple times (10 and 20 iterations) before tested on the test sets. Table 2 shows the performance of the three algorithms averaged over 5 runs.

All three algorithms produce results comparable to that of the expert system. Notably, the Maximum Entropy does very well in terms of recall, scoring above 0.98, which is better than the score of the expert system. Among the machine learning algorithms the SVM produces the best result. Whether the region is enclosed by the black edges is one of the best discriminators.

Table 2. The performance of the machine learning algorithms.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Precision | Recall | F-score |
| Naive Bayes | 0.908 | 0.897 | 0.889 | 0.892 |
| Max Ent (10 iter.) | 0.934 | 0.881 | 0.981 | 0.928 |
| Max Ent (20 iter.) | 0.928 | 0.863 | 0.985 | 0.920 |
| SVM | 0.943 | 0.900 | 0.973 | 0.935 |

In our semi-supervised learning experiment, the machine learning algorithms are trained on the training sets, which include samples labeled by the machine learning algorithms themselves as well as the samples manually labeled. In order to assign more weight on the manually labeled samples than on the machine labeled samples, we duplicate the manually labeled samples in the training sets.

Although the performance is not as good as that of the supervised learning, all three algorithms produce results comparable to that of the expert system. Especially the Maximum Entropy and SVM do very well in terms of recall, beating the expert system. The performance of the Naive Bayes improves as more correctly labeled training data are included in the training sets.

Since different algorithms tend to err on different samples, we also experiment with an ensemble machine learning method, in which multiple classifiers are combined to arrive at final classifications. We experiment with two variations of the ensemble method: The Ensemble AND, in which a sample is classified as positive if both of the classifiers agree that a sample is positive, and the Ensemble OR, in which a sample is classified as positive if at least one of the two classifiers agree that a sample is positive.

Among our ensemble classifiers, the Ensemble AND of the SVM and the Maximum Entropy with 20 iterations performs best with the F-score of 0.941, as shown in Table 3.

Table 3. The performance of the ensemble classifiers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Precision | Recall | F-score |
| SVM AND Max Ent (20 iter.) | 0.947 | 0.928 | 0.955 | 0.941 |

Figure 2 shows a result of our image segmentation process. All the text in the original image on the left is segmented and assigned the same color (red). All the other connected black pixels are grouped as non-text components. Note that black pixels that are very close in the original image are connected by a process called RLSA smoothing (Wong, 1982).



Figure 2. An example of segmentation of text.

Figure 3 shows some of the images produced as the result of the speech bubble detection. The first image from the left shows that all speech bubbles are correctly detected. The second image from the left is an example of the false negatives. The speech bubble in this image is missed because the edge of the bubble is broken and the gaps between the edges are more than 3 pixels wide. The third image from the left shows a case of the false positives. The lens of glasses are identified as speech bubbles because the eye contained in the lens has similar black pixel count and size to text. The fourth image from the left contains another example of the false positives. It is a false positive because the text is not contained in a speech bubble. It is identified as a speech bubble because the text is completely surrounded by edges. The machine learning system makes more mistakes than the expert system. However, the machine learning system sometimes correctly detects speech bubbles on which the expert system makes mistakes.



Figure 3. Examples of the images produced as the result of speech bubble detection.

### Conclusion

We are able to implement an expert system that detects the elements very accurately and a machine-learning system that does the same job as well as the expert system. The main disadvantage of the expert system is that the heuristics need to be modified when new samples are added. With machine learning, we let the machine learn from the examples by itself. We have demonstrated that machine learning can be successfully applied to a very difficult task, such as element detection in comic book panels, and they can do the task as well as the expert system.

### 

### Future Work

An interesting future goal is to be able to detect even more elements in each panel. If the machine could detect all the elements present in the panels, the scenes depicted in the panels could be described automatically.

**Note**

1. Readers interested in more details of our work can read our full-length paper at https://www.researchgate.net/publication/270546570\_ELEMENT\_DETECTION\_IN\_JAPANESE\_COMIC\_BOOK\_PANELS and also access our code at https://github.com/tkuboi/ComicbookElementDetection.

### References

**Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H.** (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations,* **11**(1).

**NLTK Project.** (2014). NLTK 3.0 documentation. http://www.nltk.org.

**Wong, K. Y., Casey, R. G., and Wahl, F. M.** (1982). Document Analysis System. *IBM Journal of Research and Development,* **2**(6): 647–56.