Extraction and Identification of Bar Graph Components by Automatic Epsilon Estimation

Sarunya Kanjanawattana and Masaomi Kimura

***Abstract*—Clustering is an unsupervised learning technique primarily used to analyze data. Density-based spatial clustering of applications with noise (DBSCAN) is effective for image clustering because it clusters neighbor objects that are located within a radius of an Epsilon parameter. However, identifying this parameter correctly requires expert knowledge. We propose methods to estimate Epsilon values effectively based on the density of each area wherein objects are located in order to extract graph components, such as axis descriptions (e.g., X- and Y-axis titles) and legends. We verified axis description extraction by measuring accuracy, precision, recall, and F-measure. The results indicate that the proposed automatic Epsilon estimation method is reliable. To evaluate legend extraction, we compared the proposed automatic Epsilon estimation to a method using a default Epsilon value (i.e., 0.6). The results demonstrate that the proposed method returns suitable Epsilon values. The proposed parameter estimation method is capable of handling graph component extraction effectively.**

***Index Terms*—DBSCAN, graph component extraction, parameter estimation, SVMs.**

I. Introduction

A graph can represent data visually in many different ways, e.g., bar and line graphs, and pie charts. In this study, we focus on bar graphs because they are relatively easy to interpret. Typically, the legend and axes descriptions provide helpful information, such as measurement units that clarify the relationships represented by the graph. Extracting such graph components should contribute to an intelligent system that can interpret latent information in a graph. However, such components, particularly legends, are positioned in various locations, and important graph characteristics are contours and texts. Clearly, to extract graph components is difficult for traditional methods such as spectral clustering.

Density-based spatial clustering of applications with noise (DBSCAN) is a simple data-clustering algorithm that is robust against noise [1]. We consider that DBSCAN is most suitable for the extraction of graph components. The DBSCAN algorithm requires two predefined parameters, i.e., Epsilon (ε), which specifies how close together the points must be to be considered part of a cluster, and MinPts, the minimum number of points required to form a dense region, i.e., within the ε distance. In addition to data inputs and clustering procedures, quality of result strongly depends on the values of the parameters. Therefore, determining suitable parameters is time consuming, because several tests are required to manually examine the most suitable parameter. Moreover, only experts with prior in-depth knowledge about the given dataset can estimate parameter values correctly. To mitigate this difficulty, parameter estimation methods have been proposed [2], [3].

In this study, we propose an effective method to extract graph components using DBSCAN algorithm with automatic ε estimation. The main objectives are to estimate ε with sufficient accuracy to obtain good clusters of graph images and extract the graph components (i.e., X-title, Y-title, and legend). The dataset used in this study is a collection of two-dimensional bar graphs that include X- and Y-titles and optionally a legend. We use DBSCAN because the input images contain many data points intensively packed together in some areas. DBSCAN is a very proficient algorithm when dealing with high-density images. For evaluation purposes, we conducted several measurements to demonstrate the performance of the proposed method. Further, in this paper, we compare our results to another method with a specific ε value.

II. Related Works

In our previous study [4], we introduced a method to correct Optical character recognition (OCR) errors from bar graphs using ontology, edit distance and dependency parsing. The graph component extraction procedure used in our previous study was very simple but effective. However, after observing the process and results over time, we considered that a more effective graph component extraction procedure could improve our previous method's performance, in particular when extracting legends, because bar graphs contain irrelevant parts that are inappropriate for conversion, such as a rectangle of bar images. Thus, a dependable graph component extraction method is required to enhance the quality of our previously proposed OCR error correction method.

A graph is a type of image that contains useful information. An interesting approach related to graph component extraction has been proposed by [5]. Kataria *et al*. presented a method to extract elements, such as axis titles, legends, and data points, from a two-dimensional graph automatically. They attempted to address the problem of overlapping text and data points. Another related graph component detection study has been presented [6]. Huang *et al.* emphasized to associate the recognition results of textual and graphical information in scientific graph images. They recognized text

Manuscript received June 25, 2016; revised August 12, 2016.

Sarunya Kanjanawattana is with Functional Control Systems, Shibaura Institute of Technology, 3-5-7 Koto-ku Toyosu, Tokyo 135-8548, Japan (e-mail: nb14503@shibaura-it.ac.jp).

Masaomi Kimura is with Department of Information Science and Engineering, Shibaura Institute of Technology, 3-5-7 Koto-ku Toyosu, Tokyo 135-8548, Japan (e-mail: masaomi@sic.shibaura-it.ac.jp).

DOI: 10.7763/IJCTE.2017.V9.1147

and graphical elements of an input image separately and combined them to fully understand the input image. They attempted to capture semantic meanings conveyed by scientific chart images. Although these previously proposed methods were effective for detecting graph components, they did not specify the type of component. In our proposed approach, each component type represents significant information; however, each type has a different role. For example, the X- and Y-titles indicate a data relationship, whereas the legend provides particular information about the data, e.g., data labels. Therefore, to enhance interpretation, it is necessary to identify the component type.

DBSCAN is a clustering algorithm that groups a set of objects in dense areas into a single cluster. The area is said to be dense if there are at least *MinPts* in a radius of ε. Finding an appropriate parameter value for each dataset is cumbersome even if the user has expert knowledge. To the best of our knowledge, few studies have focused on addressing such parameter estimation. Esmaelnejad *et al*. [2] proposed using the noise ratio of a dataset rather than the original ε. They replaced ε with a parameter that is simpler to estimate. However, identifying the presence of noise can be arduous. An unreliable method may result in an incorrect noise ratio that directly affects clustering performance. Smiti *et al.* [7] proposed an efficient clustering technique that combined the DBSCAN and Gaussian-Means (GMeans) algorithms. They used GMeans to partition data into K clusters without a predefined parameter. Then, they took the average distances between the centers of all clusters to compute a possible ε. The major drawback of this approach is that clustering was performed for both the GMeans and DBSCAN algorithms. We understand that they applied GMeans to estimate ε; however, it is unnecessary to perform the DBSCAN after GMeans to cluster data.

III. Methodology

We have separated the proposed method into two parts, i.e., axis description extraction and legend extraction. Note that parameter estimation is included in the legend extraction component.

A. Axis Description Extraction

Axis description extraction is based on the actual location of the axis descriptions. Typically, the X-title is positioned at the bottom of the X-axis. Similarly, the Y-title is typically positioned near the Y-axis, usually on the left side of the graph. To obtain the X-axis description, we partitioned the graph images downward and selected the last partition. For the Y-axis description, we also partitioned the graph from left to right and selected the first partition.

However, the initial results obtained from the above process can include irrelevant objects, such as a part of a bar and some numeric values, as illustrated in Fig. 1a. To address this problem, we integrated a pixel projection method into our system to eliminate irrelevant parts from prior results (Fig. 1b). For the X-title, after performing pixel projection in the horizontal direction, we investigated where the peaks were located. The height of the peaks denotes how many points exist along the horizontal direction. We neglected the first peak and retained the rest because the first peak often represents an internal part of a bar. The approach was similar for the Y-title; however, we retained the first peak and discarded the rest. Finally, we acquired cleaned X- and Y-titles.

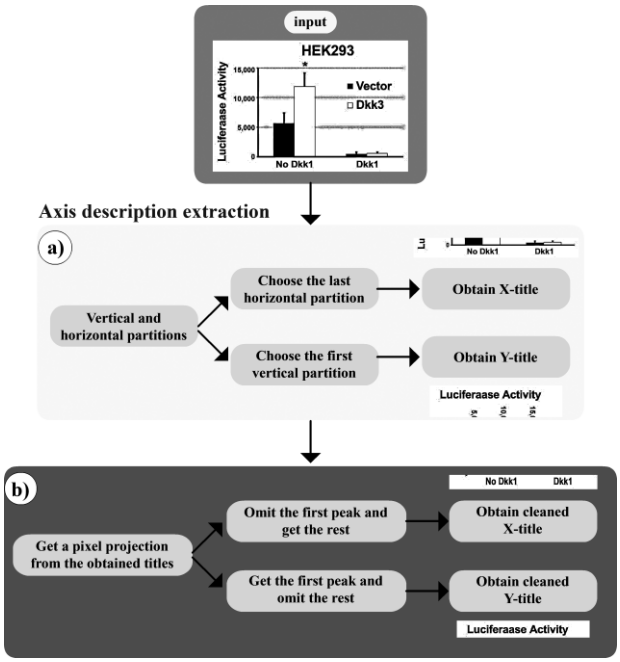

Fig. 1. Process to extract X- and Y-titles from graph images based on their location: (a) image partitioning process; (b) pixel projection.

B. Legend Extraction

A legend is a component of a graph that presents data labels. Generally, the legend optional, and its position is unfixed; thus, a method to extract a legend is more complex than axis description extraction.

Fig. 2 shows a procedure of legend extraction that presents results of each step. We divided this part into five steps: data preprocessing, data transformation, clustering, discrete Fourier transformation (DFT), and classification.

It is very important to preprocess our data because it helps to improve data quality. This process is based on the fact that the legend is typically located at the top-right of the graph rather than the bottom-left. Because we wish to eliminate as many irrelevant parts as possible, we remove axis titles. Moreover, we divided the graph into four quarters: Q1 (top-right), Q2 (top-left), Q3 (bottom-right), and Q4 (bottom-left). After considering the generality of legend position, we omit the Q4 because the legends are never displayed in this area. Finally, we obtain new inputs for our system. Fig. 2a illustrates the data-preprocessing step.

The second step is data transformation. After we obtain the new input images, it is difficult for existing systems to process them directly. We needed to transform them to a reasonable data format, such as numeric data. However, since the resolutions of the input images are quite high, we needed to rescale them to smaller sizes using average subsampling, which is a well-known technique that takes the average of a box of pixels. We employed this technique because it is fast and simple. Moreover, the rescaled results also retain the required information. We then transform them to numeric datasets. An instance of the dataset is represented by

XY-coordinates where data points are located in the given graph. Eventually, we acquired numeric datasets for each graph. Fig. 2b shows the procedures of this step.

In the clustering step, we applied DBSCAN to the numeric datasets to cluster the data based on their densities, and we cropped the graph corresponding to the clustering results, as shown in Fig. 2c. Commonly, DBSCAN always requires two parameters, i.e., ε and *MinPts*. The value of *MinPts* is a constant, and ε of the corresponding datasets should be assigned automatically. We designed the method for ε estimation based on the empirical fact that, in order to separate objects using DBSCAN clustering, we must find the shortest distance that would keep them separated. The target of this study is to detect the specific area containing the legend and extract it from the graph. As mentioned previously, it is usually located at the top-right side of the graph. To introduce our idea systematically, we distinguish five minor steps.

First, we define a squared window with odd number rows and columns that is used to identify the densities of each shifting area of each quarter. Second, after obtaining the densities, the system must find the highest density of each quarter. However, it is possible to obtain many areas that equally contain the highest density in each quarter. In this case, we repeat the density calculation with a larger window and apply it to the resulting areas to determine the new highest density. This operation is continued until we can identify the last final area. Third, when we obtain the highest density area of each quarter, we must set the center of the window as the center of the selected area. Then, we search the furthest point in the same area, which is measured by the Euclidean distance between the center and other data points of the area to be a representative of the area in each quarter. Fourth, we calculate the Euclidean distance among the obtained representatives. Fig. 3 shows the distance measurement for each quarter. Finally, we select the shortest distance and obtain ε by dividing this distance by the image width. However, there is an exception by which the system cannot obtain any density value in some quarters. We solved this problem by defining a default value for ε. If we obtain only a single cluster using the default ε, we steadily decrease the value until we acquire some clusters.

After completing clustering, we crop the graph image corresponding to the clustering result. Note that we use the rescaling method. Thus, when we return the resolutions to the original size for cropping, the scale may be different than the original image. To address this problem, we add a margin with a constant value to expand the leftmost and rightmost clusters.

The fourth step is the DFT process. After the previous step, we have obtained several cropped images, including both relevant and irrelevant results. To accomplish our objectives, we must identify a relevant part comprising the legend. To support a classification process, we apply two-dimensional-DFT (2D-DFT) to the cropped images to reduce image features in order to expose some dominant characteristics in the frequency domain (Fig. 2d). With DFT, the image is transformed to its frequency domain. We observed that the characteristics of each quarter of the DFT image can contain similar information; thus, in order to classify the legend, we select only a single quarter as input for classification, as is shown in Fig. 2d.

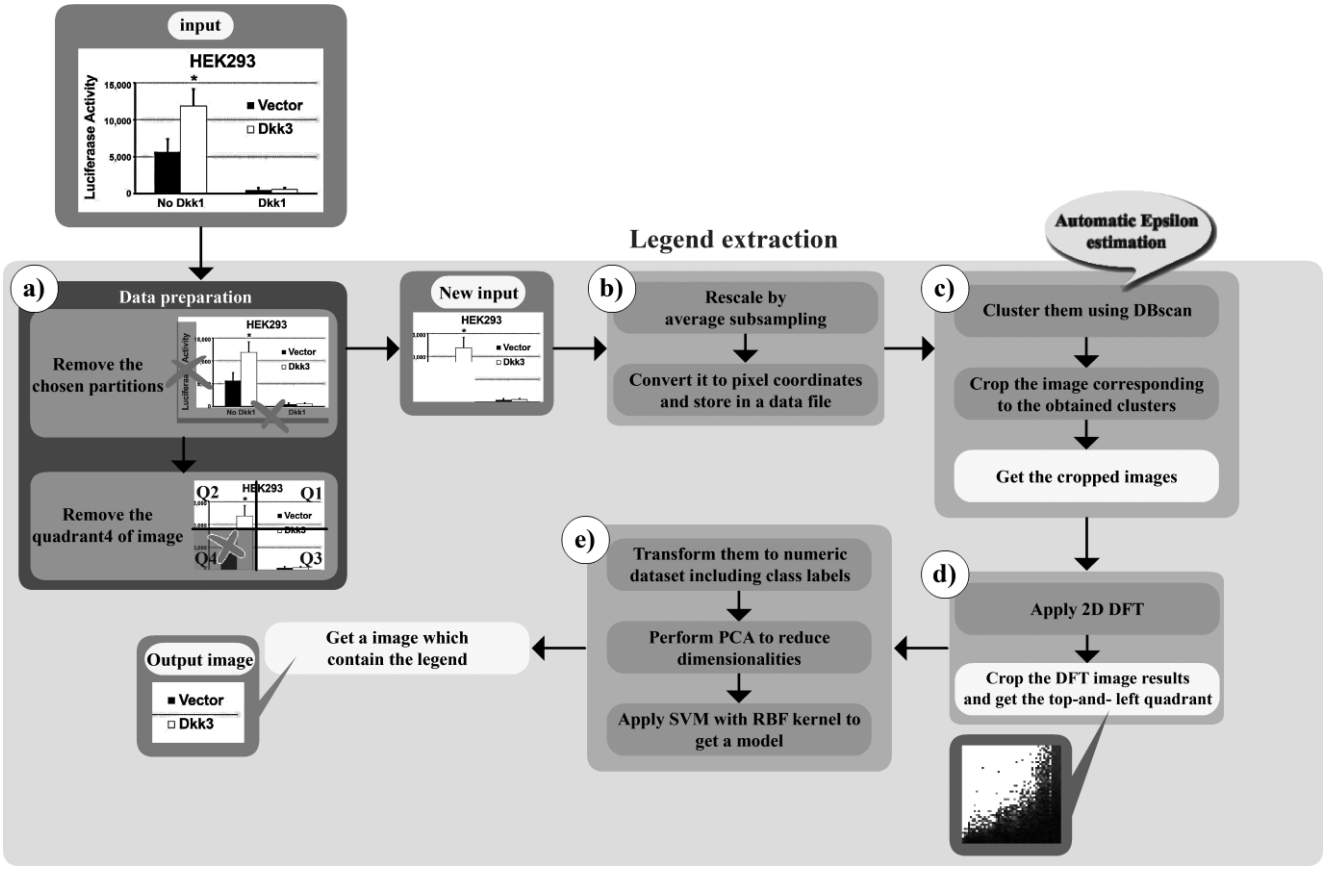

Fig. 2. Overall legend extraction procedures.

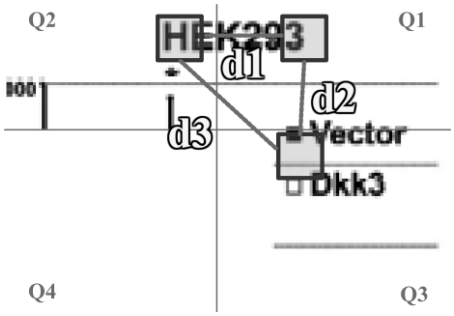

Fig. 3. ε estimation to analyze the densities of each quarter to obtain the smallest distance to be valued as epsilon.

According to analyzing the frequency of our data, the dominant characteristics obviously showed in both 1D- and 2D-DFT. Fig. 4 shows the different characteristics of DFT images based on the presence of a legend. We considered that using 1D-DFT was inadequate for our purpose because it transformed the image into frequency only along the horizontal direction that does not cover important characteristics presenting in vertical direction. To address this problem, we applied 2D-DFT to our input data because the image is analyzed in both the horizontal and vertical directions. By analyzing the horizontal direction of 1D-DFT (Fig. 4a), it can be seen that some white bands appear horizontally. These white bands represent the frequency of text. Note that most values are located in the middle- and high-frequency domains. In contrast, after observing the vertical direction, there are some changes from black to white. With the high-frequency 1D-DFT result, such changes do not occur frequently, whereas the changes in the middle-frequency domain frequently. Thus, we realized that, in 2D-DFT, the dominant characteristics should be located around the middle- to high-frequency domains. With the case shown in Fig. 4b, significant characteristics were also located in the middle- or high-frequency domains; however, the frequency patterns obviously differ. Thus, it is possible for classifying both cases separately. In our opinion, the low-frequency domain may be unnecessary for classification, particularly for classification of a legend. To obtain better results, we may omit this part by setting a threshold in order to discard it from the DFT images.

The last step is classification (Fig. 2e). We convert the results from the previous step to numeric data in order to perform classification. However, the dimensionality of our data was large, which affected classification performance significantly. Data that is too large greatly reduces performance. A simple solution is to use a dimensionality reduction technique before processing the data for classification. Here, we use principal component analysis (PCA) to emphasize the variation of each dimension and identify dominant patterns in a dataset. We analyzed the significant characteristics of the DFT images and realized that, at the middle and high 2D-DFT frequency domains, the variations were significant because there were many frequency changes. To obtain appropriate characteristics, we must retain features that contain high-frequency variation. Clearly, PCA can properly address this problem because it ranks variations and selects the top features. To perform this properly, we specify the number of desired dimensions and obtain a new numeric dataset with much lower features. Then, we apply a support vector machine (SVM) to the dataset using a radial basis function (RBF) kernel to classify the images with a legend. An SVM is a powerful technique that can handle data whose characteristics are separable by a hyperplane. Our dataset contained high dimensionality and was numeric data. Moreover, as shown in Fig. 4, the characteristics of our input data with and without a legend are clearly distinguishable. Thus, an SVM is a good candidate for our classification.

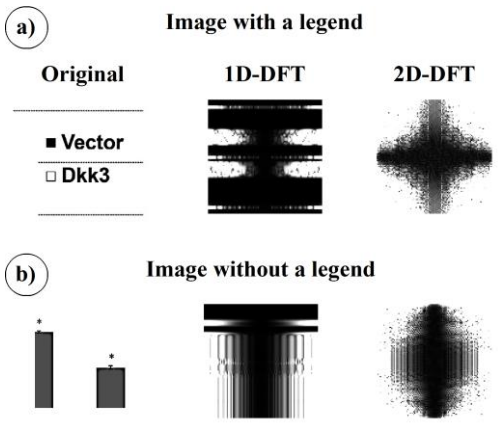

Fig. 4. Examples of DFT results that present the difference between an image (a) with and (b) without a legend.

Regarding the kernel used in this study, after analyzing the characteristics of our data, we realized that a linear kernel was inappropriate because our data cannot be separated by a linear hyperplane. As mentioned previously, the dominant characteristics of our data are located in the middle or high-frequency domains. When we transformed the DFT images to a numeric dataset, those characteristic features were scattered along a single dataset record, as illustrated in Fig. 5. According to the distribution of frequencies, it is difficult to use a linear hyperplane to separate the middle or high-frequency domains from the low-frequency domain. Based on our numeric dataset, we must use a nonlinear kernel, frequency of each image is often located at nearby features; therefore, we can use the RBF kernel to split such features using a hyperplane.

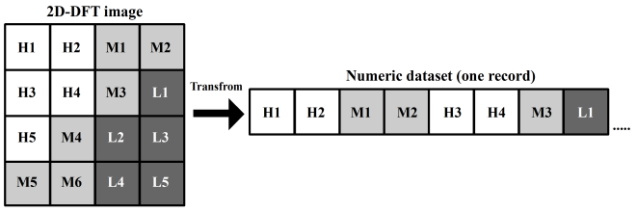

Fig. 5. Data transformation from a 2D-DFT image to a single-row numeric dataset used for classification.

IV. Experiment and Results

We conducted an experiment to evaluate whether the proposed method can automatically introduce suitable ε values to DBSCAN for effective clustering. We compared the proposed method with automatic parameter estimation (METHOD 1) to a method with a default ε value of 0.6 (METHOD 2).

Accuracy, precision, recall, and F-measure are discussed in this section. Accuracy is a statistical measurement of how well a classification test precisely identifies corrected instances. A higher accuracy rate means that the predicted values are very similar to the given values. Precision statistically presents the measurement of how many outputs are classified as positives. Recall is another statistical measurement of how well the outputs cover the positives. F-measure is an averaged combination of precision and recall.

For the axis description extraction, we extracted the X- and Y-titles from the input images. We manually evaluated the extraction results and verified the number of obtainable and the number of relevant axis-titles. Fig. 6 shows the accuracy, precision, recall, and F-measure results. Obviously, the proposed method effectively extracted the axis titles. However, after considering relevant results, the precision of both titles was reduced slightly compared to the accuracy rates because some images contained unnecessary parts (such as part of a bar). Meanwhile, the recall rates were remarkable.

For legend extraction, we analyzed both legend identification and classification. Legend identification identifies and extracts the legend from the graph. Here, legend classification indicates how well the outputs are classified as a legend. Note that there were 54 images containing a legend in our dataset. After checking manually, the results of METHOD 1 indicated that this method gave correct clusters that can be used to properly detect the legend (identification rate of 93%). METHOD 2 gave some unsatisfactory results and a low identification rate, i.e., approximately 31%. We evaluated legend classification using a 50% split of the dataset, i.e., one half was used to create a model and the other half was used to test the model. Typically, an SVM with an RBF kernel requires two parameters, i.e., *cost* and *margin* (γ). To define optimal SVM parameters, we utilized a grid search [8]. For our dataset, we defined the *cost* and γ as 2 and 0.00049, respectively. The accuracy rate of METHOD 1 was 93%, and the accuracy rate of METHOD 2 was 83%. Clearly, METHOD 1 provided much higher accuracy than METHOD 2.

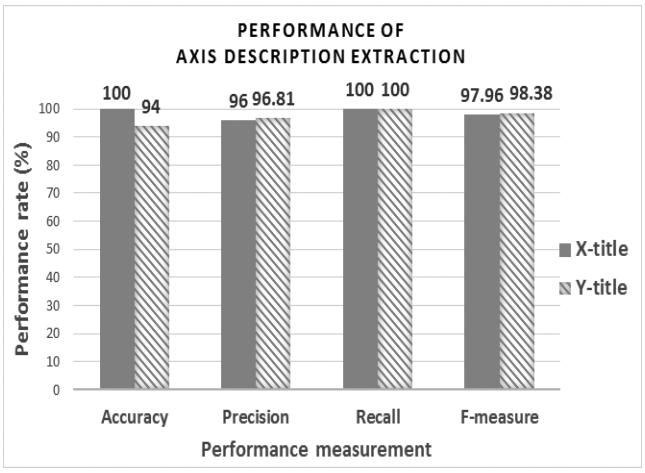

Fig. 6. Performance rates of axis description extraction.

Table I shows the results of METHOD 1. Note that the class label "Yes" represents correctly classified images containing a legend.

TABLE I: EVALUATION RESULTS OF CLASSIFICATION FOR METHOD 1

Method	Accuracy	Precision "Yes"	Recall "Yes"	F-measure "Yes"
METHOD 1	93%	79%	82%	81%

V. DISCUSSION

We have focused on developing a method to extract graph components and have introduced an effective method to automatically identify ε values for DBSCAN. This study contributes to our earlier research [4] in an effective manner. In an experiment, we tested the performance of the proposed method by observing the number of axis descriptions and legends that were extracted correctly. Moreover, automatic ε estimation was evaluated by comparing another method using a default ε value of 0.6. The input used in this study was a collection of bar graphs (100 images; 54 images with a legend).

The accuracy, precision, and recall of our results for axis description extraction were greater than 90%, as shown in Fig. 6, because the extraction process for the axis description is based on the nature of a graph's structure. The X-title is always located at the bottom of graphs, whereas the Y-title is at the left side. Therefore, our idea is effective to achieve our objective However, the accuracy rate of Y-title extraction decreased trivially compared to X-title extraction because image noise led our system misunderstand to select a wrong position to cut a peak. Thus, the error results (e.g., incomplete titles) were presented occasionally. The precision rates were insignificantly lower than the accuracy rates because we found some results that still contained unnecessary parts, even if we had already cleaned them. Such errors were caused by image noise. Note that the relevant result represents the extracted title containing only text, not included the irrelevant parts.

METHOD 1 provided a very high identification rate (93%) compared to METHOD 2. The identification rate of METHOD 2 was low (31%) because most outputs were original images that contained many irrelevant parts. Our target is only the part of the image with a legend. The clustering of METHOD 2 could not provide good results. However, METHOD 1 is suitable for extraction of a legend from a graph.

Regarding legend classification accuracy, METHOD 1 provided very high accuracy compared to METHOD 2 because the proposed method (i.e., METHOD 1) introduced suitable ε parameters for clustering relative to the input data (METHOD 2 used a default ε value). However, the recall of class "Yes" was slightly low because some graphs with a legend were misclassified. After observing that the obtained results caused these errors, we found that the images contained the legend including irrelevant parts. Thus, when using 2D-DFT, most dominant characteristics came from unrelated areas rather than the legend, which negatively affected classification performance.

VI. CONCLUSION

We have proposed a graph component extraction method

that uses DBSCAN, 2D-DFT, and an SVM. We have also introduced automatic ε estimation to obtain a suitable parameter value for each input image. We conducted an experiment to evaluate the performance of the proposed method by comparing it to another method that used a default ε value. We have investigated an effective idea that can be applied to our previously proposed method [4].

The accuracy, precision, and recall rates of the proposed axis description extraction were greater than 90%. We measured accuracy by checking the number of correct and complete components obtained by our system. However, due to the presence of irrelevant parts, the precision rates were slightly reduced comparing to other rates. For legend extraction, we have discussed the quality of the proposed method using suitable ε. Typically, this has been a problem in many previous studies that cannot identify the true ε relative to their data. As a result, we conclude that the proposed method can provide reasonable performance comparing to METHOD 2. The accuracy rate of the proposed method was up to 93%, and the precision rates were greater than 80%.

In future, we will combine the methods proposed in this study to improve our previously proposed OCR error correction method.

REFERENCES

[1] Q. Ye, W. Gao, and W. Zeng, "Color image segmentation using density-based clustering," in *Proc. 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. III-345-348, 2003.

[2] J. Esmaelnejad, J. Habibi, and S. H. Yeganeh, "A novel method to find appropriate ε for dbscan," in *Proc. Intelligent Information and Database Systems, Springer*, pp. 93-102, 2010.

[3] A. Karami and R. Johansson, "Choosing dbscan parameters automatically using differential evolution," *International Journal of Computer Applications*, vol. 91, no. 7, pp. 1-11, 2014.

[4] S. Kanjanawattana and M. Kimura, "A novel method of OCR-error correction applied to bar graphs based on ontologies," 2016.

[5] S. Kataria, W. Browuer, P. Mitra, and C. L. Giles. (2008). Automatic extraction of data points and text blocks from 2-dimensional plots in digital documents. In *Proc. the 23rd National Conference on Artificial Intelligence - Volume 2*. [Online]. pp. 1169-1174. Available: http://dl.acm.org/citation.cfm?id=1620163.1620254

[6] W. Huang, C. L. Tan, and W. K. Leow, "Associating text and graphics for scientific chart understanding," in *Proc. Eighth International Conference on Document Analysis and Recognition*, pp. 580-584, 2005.

[7] A. Smiti and Z. Elouedi, "Dbscan-gm: An improved clustering method based on gaussian means and dbscan techniques," in *Proc. 2012 IEEE 16th International Conference on Intelligent Engineering Systems*, pp. 573-578, 2012.

[8] A. A. Salatino. (February 2014). Grid search svm. [Online]. Available: http://infernusweb.altervista.org/wp/?p=786

Sarunya Kanjanawattana was born in Naknonratchasima, Thailand, in 1986. She received the B.E. degree in computer engineering from Suranaree University of Technology, Nakhonratchasima, Thailand, in 2008, and M. Eng from Asian Institute of Technology, Pathum Thani, Thailand in 2011. Currently, she is a Ph.D. student of Shibaura Institute of Technology, Tokyo, Japan.

In 2011, she joined National Electronics and Computer Technology Center, Thailand, as a research assistance. Her project was related to find an optimal solution of traffic congestion. In the present, she works at Suranaree University of Technology as a lecturer in the department of computer engineering. Her research interests included data mining, machine learning, natural language processing, ontology and image processing.

Masaomi Kimura received a B.E. (1994) degree in precision engineering, and M.S. (1996), and D.S. (1999) degrees in physics from University of Tokyo. He is a professor, department of information science and engineering, Shibaura Institute of Technology. His current interests include data engineering, complex systems and medical safety.