Image Processing and Image Mining using Decision Trees*

KUN-CHE LU AND DON-LIN YANG
Department of Information Engineering and Computer Science
Feng Chia University
Taichung, 407 Taiwan

Valuable information can be hidden in images, however, few research discuss data mining on them. In this paper, we propose a general framework based on the decision tree for mining and processing image data. Pixel-wised image features were extracted and transformed into a database-like table which allows various data mining algorithms to make explorations on it. Each tuple of the transformed table has a feature descriptor formed by a set of features in conjunction with the target label of a particular pixel. With the label feature, we can adopt the decision tree induction to realize relationships between attributes and the target label from image pixels, and to construct a model for pixel-wised image processing according to a given training image dataset. Both experimental and theoretical analyses were performed in this study. Their results show that the proposed model can be very efficient and effective for image processing and image mining. It is anticipated that by using the proposed model, various existing data mining and image processing methods could be worked on together in different ways. Our model can also be used to create new image processing methodologies, refine existing image processing methods, or act as a powerful image filter.

Keywords: data mining, pixel classification, low level image processing, image mining, decision tree

1. INTRODUCTION

Mining large datasets of different types is a future trend [1]. As computer technologies become more ubiquitous, besides numerical and categorical data, various digitalized images, sounds, voices, and videos have become part of daily life. Yet, although plenty of knowledge can be hidden in these data, very little literature has discussed data mining on them.

Image mining concerns the extraction of implicit knowledge, image data relationship, or other patterns not explicitly stored in the images [2]. It is more than just an extension of data mining to image domain. Image mining is an interdisciplinary endeavor which draws upon expertise in computer vision, image understanding, data mining, machine learning, database, and artificial intelligence [3]. In [4, 5], the authors have classified the issues of image mining into four classes:

1. association
2. classification
3. sequential pattern
4. time series pattern.
Vailaya et al. [6] used low-level visual features to capture high-level concepts. They performed a hierarchical classification on vacation images to classify them as indoor or outdoor. Outdoor images were further classified as city or landscape; and a subset of landscape images was classified into sunset, forest, and mountain classes. Cromp et al. [7] examined various classifiers at pixel-level for automatic content extraction of remotely sensed images. No claims on superiority of one classifier over another were made because results depended on the situation. In [8], a new type of texture feature based on association rules is proposed. It is shown that association rules can be adapted to capture frequently occurring low-level structures in images. Other research on developments in image mining such as techniques, particular frameworks, and systems has been surveyed in [2].

Image mining is just at its infancy [9]. Most existing data mining techniques have been designed for mining categorical or numerical data and are not well suited for image mining. On the other hand, various image processing techniques such as image segmentation, image enhancement, image restoration, or image compression focus on manipulation, not on analyzing image data. Thus, although many image processing techniques have been developed, few of them can be adopted to mine image data.

In image retrieval [10, 11], data mining techniques may be used on metadata and/or content of images for indexing an image. Yet, the primary concern of image retrieval is to effectively retrieve similar image objects under a user specified query. On the whole, there has been relatively little concern in extracting useful knowledge and/or processing the indexed images until recently.

Due to most existing technologies that cannot be directly applied for image mining, in this study, we tried to solve the problem by presenting a novel approach based on the decision tree for both image mining and image processing. The decision tree induction is a well-known methodology used widely in various domains, such as artificial intelligence, machine learning, data mining, and pattern recognition [12, 13]. It is a predictive model which usually operates as a classifier. The construction of a decision tree, also called decision tree learning process, uses a training dataset consisting of records with their corresponding features and a label attribute to learn the classifier. Once the tree is built, it can be used to predict the label attribute of unidentified records that are from the same domain. A decision tree has the hierarchical form of a tree [14]; an example is shown in Fig. 7.

There are many advantages to the use of decision trees for classification [15]. Decision trees certainly are easy to use and efficient. Rules can be generated that are easy to interpret and understand. They scale well for large databases because the tree size is independent of the database size. Each tuple in the database must be filtered through the tree. This classification takes time proportional to the height of the tree, which is fixed. Moreover, trees can be constructed for data with many attributes.

In our framework, the image used for input is formatted as a set of equally sized raw and label image pairs. These two kinds of images can be considered as the images “before” and “after” an ideal processing or the images “after” and “before” a corruption. With decision tree induction, the proposed model can be used to mine hidden relationships between attributes and the target label from image pixels; it can also select important features. Image processes or image effects can be modeled as understandable decision rules. Moreover, these rules can be applied to perform pixel-wised image processing.
on new images. Based on different local pixel information, different image processing facilities can be applied easily at different image locations within a single scan of the test image. The built image processing model is efficient and effective, and can be more sophisticated than other existing image processing methods.

2. PROPOSED MODEL

Fig. 1 depicts a general flow chart of the proposed image mining and image processing framework. Each part will be described in the following subsections.

![Image Flow Chart]

Fig. 1. General processing flow of the proposed framework.

2.1 Image Transformation and Feature Extraction

As mentioned, the input data of the proposed model is formatted as a set of equal sized raw and label image pairs. The transformation of the input image dataset into a database-like table and subsuming of the related features is described in this subsection. For the sake of clarity, various terms used for this process are defined below. In addition, we propose three kinds of input data sources, which will be discussed further in section 4.1.

**Definition 1** The raw image is a $d$-dimensional light-intensity function, denoted by $R(c_1, c_2, \ldots, c_d)$, where the amplitude (or value) of $R$ at spatial coordinates $(c_1, c_2, \ldots, c_d)$ gives the intensity of the raw image at that point (or pixel).

**Definition 2** The label image is a $d$-dimensional light-intensity function, denoted by $L(c_1, c_2, \ldots, c_d)$, where the value of $L$ at spatial coordinates $(c_1, c_2, \ldots, c_d)$ gives the class identifier of the pixel at same spatial coordinates of its corresponding raw image.

**Definition 3** The database-like table $X = \{x_1, x_2, \ldots, x_t\}$ is a set of records, where each record $x_i \in \mathbb{R}^k$ is a vector with elements $<a_1, a_2, \ldots, a_k>$ being the value of attributes (or features) of $X$. 
In this work, only $d = 2$ is considered, i.e., images with dimensionality of 2. An example for an atom of the input image dataset is shown in Fig. 2. Each pixel value of the raw image represents the gray level of a pixel. Each pixel value of the label image represents the class label of the pixel. Both pixel values are in the same position.

In this example, the raw image contains the capital English letter “I” with certain degree of blur. Thus, the inside pixels of the letter are darker and the outside pixels are brighter. If a pixel in the label image has the value “1”, the pixel in the same position of the raw image is a pixel of outside contour. It is assumed to be a pixel of interest (POI) in this case. In practice, the pixel value of the label image is not limited to the binary form but could take any kind of form. In addition, we can have as many raw and label image pairs at the same time as required for the input.

In order to mine useful information from a set of raw and label images, we propose a methodology to transform them into a database-like table and allow any data mining algorithms to work on top of the table. This process is simple and straightforward as shown in Fig. 3. Fig. 4 shows a part of the results of this transformation process according to the data in Fig. 2. Each row of such result table stands for a pixel. Hence its cardinality (number of rows) equals the number of total pixels in the raw image. In addition, each column of such table represents a feature associated with the given pixels.

```plaintext
procedure img2tab(image: raw, label);
begin
set feature_generated_functions[1..n];
set label_generated_function;
initiate table, pixel;
while pixel exists do
{pixel scanning process}
insert into table value :=
  feature_generated_i(raw, pixel),
  ...
  feature_generated_n(raw, pixel),
  label_generated(label, pixel);
continue to scan on the next pixel;
end while
return table;
end
```

Fig. 4. Part of the result of the transformation process.

Fig. 3. Pseudocode of the image transformation algorithm.
feature$_1$ represents the gray level and feature$_2$ the local variation [16]. In order to simplify this demonstration, the local variation in this case is replaced with the average difference of a pixel to its 4-neighbors. Other pixel-wised features [17, 18] such as entropy, contrast, mean, etc. can also be encoded into the table as long as they might have affection on the collected dataset.

Various encoding strategies such as normalization (e.g., adjusting the value ranging from 0 to 1) or generalization (e.g., transforming the value to high, medium, or low) can be applied when generating the desired features. Moreover, the label image was included as a column in that table. With the presence of the label feature, hidden relationships between these two kinds of images can be mined.

### 2.2 Data Reduction

Because of the image characteristics, pixels from a neighboring area will generate similar feature vectors in the transformation process. Under some circumstances, it will cause remarkable redundant information in the result table; for example, an image with a large portion of background. Here we present some basic types of redundancy and show how they can be eliminated while converting the input image set.

**Definition 4** The feature scope of a pixel M with spatial coordinates $(c_1, c_2)$ is an $n \times n$ pixel area with center at M, from which all the desired features of M can be generated. Usually $n$ is an odd number, and the sub-image within the feature scope, i.e., pixels within spatial coordinates $(c_1 \pm \frac{n-1}{2}, c_2 \pm \frac{n-1}{2})$, is called the root space of the pixel M, denoted as $\{RSM\}$.

**Definition 5** Two root spaces $\{RS_N\}, \{RS_O\}$ are rotation reachable if $\{RS_N\} = \{RS_O\}^R$, where $\{\}_{R}$ stands for a root space after rotating the angle once by 90°, 180°, or 270°.

**Definition 6** Two root spaces $\{RS_N\}, \{RS_O\}$ are mirror reachable if $\{RS_N\} = \{RS_O\}^F$, where $\{\}_{F}$ stands for a root space after flipping horizontally or vertically.

Given two pixels $P$ and $Q$ at different spatial coordinates of an image $I$, they are said to be:

1. **equivalent redundant**, if $\{RS_P\}$ is equal to $\{RS_Q\}$,
2. **rotation redundant**, if $\{RS_P\}$ and $\{RS_Q\}$ are rotation reachable,
3. **mirror redundant**, if $\{RS_P\}$ and $\{RS_Q\}$ are mirror reachable,
4. **conflict redundant**, if $\{RS_P\}$ and $\{RS_Q\}$ satisfy any one of the first three conditions, but the label information of pixels $P$ and $Q$ is not equal to each other.

<table>
<thead>
<tr>
<th>pixel$_1$</th>
<th>feature$_1$</th>
<th>feature$_2$</th>
<th>...</th>
<th>feature$_n$</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>pixel$_2$</td>
<td>9</td>
<td>1.25</td>
<td>...</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>pixel$_3$</td>
<td>9</td>
<td>0</td>
<td>...</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>pixel$_4$</td>
<td>7</td>
<td>2</td>
<td>...</td>
<td>value$_{2,5}$</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig. 4. Result table of image transformation according to the input in Fig. 2.
function \( RR \) (image: raw, label: pixel; C);
begin
apply quantization on \( \{RS_C\} \) if necessary;
if \( \{RS_C\} \) can be matched in \( \Xi \) do
{redundant pixel}
discard \( \{RS_C\} \) for further record generation;
if the label information of the two matched entries are not
equal do (conflict redundant pixel)
update the corresponding information in \( \Xi \);
retrieve or update previously generated record if necessary;
else
{non-redundant pixel}
record all characterized redundancies of \( \{RS_C\} \) and
the corresponding label information in \( \Xi \);
end

Fig. 5. Pseudocode of the redundancy reduction algorithm.

1. \( \{RS_1\} = 799 \) 2. \( \{RS_2\} = 799 \) 3. \( \{RS_3\} = 799 \) 4. \( \{RS_4\} = 699 \) 5. \( \{RS_5\} = 699 \)
add \( \langle\{RS_1\},1\rangle\) to \( \Xi \) add \( \langle\{RS_2\},0\rangle\) to \( \Xi \) add \( \langle\{RS_3\},0\rangle\) to \( \Xi \) discard \( \langle\{RS_4\},0\rangle\) discard \( \langle\{RS_5\},1\rangle\)

6. \( \{RS_6\} = 799 \) 7. \( \{RS_7\} = 799 \) 8. \( \{RS_8\} = 799 \) 9. \( \{RS_9\} = 799 \) 10. \( \{RS_{10}\} = 799 \)
add \( \langle\{RS_6\},1\rangle\) to \( \Xi \) add \( \langle\{RS_7\},1\rangle\) to \( \Xi \) add \( \langle\{RS_8\},0\rangle\) to \( \Xi \) discard \( \langle\{RS_9\},1\rangle\) discard \( \langle\{RS_{10}\},1\rangle\)

11. \( \{RS_{11}\} = 579 \) 12. \( \{RS_{12}\} = 579 \) 13. \( \{RS_{13}\} = 579 \) 14. \( \{RS_{14}\} = 579 \) 15. \( \{RS_{15}\} = 579 \)
add \( \langle\{RS_{11}\},0\rangle\) to \( \Xi \) add \( \langle\{RS_{12}\},1\rangle\) to \( \Xi \) add \( \langle\{RS_{13}\},0\rangle\) to \( \Xi \) discard \( \langle\{RS_{14}\},1\rangle\) discard \( \langle\{RS_{15}\},1\rangle\)

16. \( \{RS_{16}\} = 079 \) 17. \( \{RS_{17}\} = 079 \) 18. \( \{RS_{18}\} = 079 \) 19. \( \{RS_{19}\} = 079 \) 20. \( \{RS_{20}\} = 079 \)
add \( \langle\{RS_{16}\},1\rangle\) to \( \Xi \) add \( \langle\{RS_{17}\},0\rangle\) to \( \Xi \) add \( \langle\{RS_{18}\},0\rangle\) to \( \Xi \) discard \( \langle\{RS_{19}\},1\rangle\) discard \( \langle\{RS_{20}\},1\rangle\)

21. \( \{RS_{21}\} = 579 \) 22. \( \{RS_{22}\} = 579 \) 23. \( \{RS_{23}\} = 579 \) 24. \( \{RS_{24}\} = 579 \) 25. \( \{RS_{25}\} = 579 \)
add \( \langle\{RS_{21}\},1\rangle\) to \( \Xi \) add \( \langle\{RS_{22}\},0\rangle\) to \( \Xi \) add \( \langle\{RS_{23}\},0\rangle\) to \( \Xi \) discard \( \langle\{RS_{24}\},0\rangle\) discard \( \langle\{RS_{25}\},1\rangle\)

Fig. 6. Result of the redundancy reduction algorithm according to the images in Fig. 2.
Users could characterize other types of redundancy according to the image problem they wish to solve. In order to pinch more redundancies, quantization techniques can be applied on the root space. The pseudocode regarding the function of redundancy reduction is shown in Fig. 5. This function can be added to the pixel scanning process of the image transformation algorithm in Fig. 3. Fig. 6 shows the results of this reduction process according to the images in Fig. 2. The number of pixels for transformation after reduction has reduced from 25 to 9.

2.3 Mining Results and their Applications

After having obtained such a database-like table in accordance to the desired input image dataset, mining algorithms can then be used on it. In this study, we have chosen the decision tree for this purpose. An advantage of the decision tree over other methodologies, such as neural networks, is that it can provide understandable English-like rules or logic statements. For instance, if the gray level of a given pixel ranges between 180 and 240 and its entropy is greater than 0.5, then it is a pixel of interest, POI. This basic idea of simplicity and easy understandability is also the main principle of our approach.

The results of such a mining process may help us to better understand the image properties and relate to real world instances. The results can also be used to process new images of the same domain. Basically, the result of the proposed model is a decision-tree classifier. Fig. 7 depicts a classifier derived from the data shown in Fig. 4 by using CART [15]. A result classifier can be further straightforwardly translated into a set of human-readable if-then rules. For instance, from the three leaf nodes in Fig. 7, we can obtain the following three rules:

− If the gray level of a given pixel is less than 8 and its local variation is less than 5, then it is a pixel of outside contour.
− If the gray level of a given pixel is less than 8 and its local variation is greater than or equal to 5, then it is not a pixel of outside contour.
− If the gray level of a given pixel is greater than or equal to 8, then it is not a pixel of outside contour.

![Decision Tree](image)

Fig. 7. A decision tree for the concept is outside contour, derived from Fig. 2, indicating whether or not a pixel is a pixel of outside contour.
These rules can provide useful information about the training image. Besides, in order to obtain a higher level of appearance and meet the different information granularity requirements, the rules can be post-processed by rule induction algorithms [19]. More prominently, they can be used to process new images from the same domain. The practical image processing capabilities include image restoration, image enhancement, image segmentation, etc. Both experimental and theoretical analyses were performed in this study to examine the proposed model.

The built classifier can also be used to select important features. Features used at higher tree levels for the splitting criteria show a higher significant influence on the pixel class. The selected features can reflect the characteristics of the label image and help design or refine other image processing algorithms.

3. EXPERIMENTS

For our mining process, we used the decision tree package of SAS Enterprise Miner (EM) [20]. First, we performed image restoration with enhancement by using two types of synthetic images, one by Arabic numerals and the other by English alphabet, with added noise artifacts. Afterwards, we performed image segmentation by using the same types of synthetic images with added blur artifacts. In all our experiments, the synthetic images were created with Photoshop by using 24-point Verdana fonts with a smooth boundary. The noise artifact was generated by using the filter, add_noise, with a parameter of uniform distribution and the amount of 100%. Another blur artifact was generated by using filter, Gaussian blur, with a radius of 2.5 pixels.

In the training dataset of Arabic numerals, the distorted and the original images synthesized by the numerals “3” to “9” and their two combinations (i.e., 33, 34, …, 98, 99) were used for the raw and label images, respectively. Fig. 8 shows the training dataset of the case with noises. The other numerals (i.e., 0, 1, and 2) were used to synthesize the testing image dataset.

(a) Raw image.                   (b) Label image.
Fig. 8. Syntactic training image dataset of the noised numerals.

Similarly, in the English alphabet training dataset, the distorted and the original images synthesized by the letters “F” to “Z” and their two combinations (i.e., FF, FG, …, YZ, ZZ) were used for the raw and label images, respectively. The other letters (i.e., A, B, …, E) were used to synthesize the testing image dataset.
For image transformation, a feature scope of size $5 \times 5$ was used and the selected features included gray level, local variation, mean, local minimum, local maximum, and entropy. The label of a given pixel in the experiments of image restoration with enhancement was set to its gray level in the label image. We did not apply any encoding strategies mentioned in section 2.1 on the features to simplify the demonstrations. However, in practice, we can use any encoding strategy if required. In the image segmentation experiments, the label feature was transformed to 0 or 1 according to the thresholded label image. In this way the segmentation nature was imitated to distinguish between “background” or “object”.

After we have settled the transformation details, a database-like table can be derived. By applying a classification algorithm on the database-like table, a classifier for label prediction can be obtained. Under the same way, testing images can be transformed into a database-like table to predict the label attributes. These predicted labels can moreover be visualized in a natural form of the input data, i.e., image. As we are proposing a general image mining and image processing framework and any existing decision tree algorithms can be used to do the job, we show only the testing result to simplify the demonstration. For the other results regarding the constructed classifier or the corresponding rules, if interested, examples can be found in our previous work [21].

![Fig. 9. Results of image restoration with enhancement; top row: original images (ground truth); middle row: distorted/testing images; third row: restored/resulting images.](image)

![Fig. 10. Results of image segmentation; top row: original images (ground truth); middle row: distorted/testing images; third row: restored/resulting images.](image)

Figs. 9 and 10 show the testing images along with the testing results and the ground truth from the experiments for image restoration with enhancement and image segmentation, respectively. The results in Fig. 9 show that the noised images were successfully restored and enhanced. The results in Fig. 10 show that the main portions of the image object were successfully segmented.

Fig. 11 compares the testing results of the proposed method and the thresholding technique. The first row in each of the examples were obtained by the proposed method and the others by the thresholding with different thresholds. The numerical testing results in Fig. 11 (a) were obtained by using the alphabet training dataset, which were better
than those obtained by using the numerical training dataset shown in Fig. 9. The reason is that there were only 56 training samples in the numerical training case, while the English alphabet training case had 462 samples. The results in Fig. 11 show that our method can easily produce better results than the thresholding.

We also assessed the built model by using blurred and noised images acquired from a scanner. Fig. 12 shows parts of these two kinds of images that were used in our experiment. The blur phenomenon may be caused by the gutter and/or movement of the scanning material. The noise phenomenon may be caused by scanning crumpled and/or grained paper.

In this experiment, both the English alphabet and Arabic numerals were used to synthesize the training image, where synthetic blurred and noise phenomena were added jointly to imitate real cases. The built image processing model was applied to both kinds of the real test images. OCR software was adopted to recognize the characters, and the numbers of characters erroneously recognized were used to evaluate the quality of the resulting images. We also compared the built model with the thresholding technique, where the threshold value was set automatically to 128. The results in Fig. 13 show that the built model is much more effective.

4. THEORETICAL ANALYSES AND DISCUSSIONS

4.1 The Input Data Source

Various image processing capabilities of the proposed model can be achieved by manipulating the input data source. As previously mentioned, there are three different sources for the label image:
4.1.1 User defined

With the advent of computer technology, more and more images will be presented in an electronic form. Hence, we will need more advanced methods to deal with these images as they are becoming an indispensable part of our daily life. Users would expect to obtain the effect according to their specification after processing an image. However, suitable processing methods may not always be available. In that case, the users or domain experts can artificially manipulate the label image based on their requirements. By applying our approach, a desired image processing method can be acquired. Moreover, the mined information and selected features may help better understand the properties of the desired effects. This understanding also helps to design or refine other related image processing algorithms.

4.1.2 From other algorithms

If certain algorithms have been developed to solve particular image processing problems, users can use the outcomes they already obtained as the label image. As a result, the built model can process images similarly to or even better than the original method. In this manner, the proposed model is able to mine, learn, and explain the decision process of the original method. This explanation capability is particular helpful when the original method is more or less a black box for humans, such as a neural network.

On the other hand, users may include all the features that might have affection on the label image in the database-like table and leave the selecting process to the decision tree. The learning process of the decision tree is comprised of the following two processes that are necessary when building image interpretation systems: feature selection and learning of the classifier [22]. This objective feature selection method can pick up more relevant features that are related to the label feature compared to the subjective feature selection methods. The mined information and selected features may help users better understand the studied properties and/or refine the original method.

4.1.3 Hybrid

The training data for the hybrid method are obtained through exploiting the outcome of other algorithms and additional modifications made by the user. In this way, the
flaw of the original method may be overcome more easily. Furthermore, our expertise can be embedded in the new model, and thus make it more sophisticated.

4.2 Efficiency and Effectiveness

Once the image model has been trained, only one scan of the image dataset is required to perform processing or testing on the new images. The processing time is proportional to the height of the result decision tree, which is fixed and independent of the original data size. The built decision tree model can also generate an efficient and compact code [23]. With these efficiency characteristics, our approach can be used to relieve the time-consuming problem of other image processing methods, such as neural networks. Two ways to utilize the proposed approach are possible. In the first one, the original method is superseded with our method by using the strategies mentioned in section 4.1. The second one is to use the built model as an efficient filter to single out those images that need extra processing. For example, after singling out suspicious mammograms that might contain pixels of cancer [24], the original method can be applied for further diagnosis.

The first option is suitable for the case when the generated method, by applying our approach, is able to process the image better than the original method. Otherwise, a slight loss of accuracy does not make a significant difference to the users [22]. The second option is suitable for the case when the processing result has a critical usage, and the generated method is unable to fulfill this requirement. When the proposed approach is used as an efficient filter, to ensure the desired types of defects can be pinpointed, the misclassification penalty for different classes can be adjusted in the decision tree training process. The capability to control a decision tree is a big advantage which can help manipulate the classification process or even interactively train our model [20]. With these abilities, the well-known classification problem, over-fitting, can be handled more easily.

4.3 Extensibility and Flexibility

Many image processing methods possibly involve the use of spatial filters (or masks). Various kinds of masks can be integrated in the built model at the pixel scanning process of the image transformation procedure. After application of a mask on the training images, the resulting gray level as well as the features of a given pixel can also be included as a column within the database-like table. Thus, the proposed model is able to mine across different layers of the image, and determine important features over disparity of the masks as shown in Fig. 14. Apart from image masks, various image processing methods can also be incorporated into the model. Based on different local pixel information, suitable image processing capabilities can be easily derived via the different rules invoked from a built model. The proposed model can also be easily extended to handle 3D, 4D, or even higher dimensionality of image without the need of a big revolution.

5. CONCLUSION AND FUTURE WORK

In this paper, a novel, efficient, and effective model was proposed. With our approach, data mining and image processing technologies can be fused. The results have
shown clearly that the proposed model can mine and process image data well. The experiments have demonstrated the adaptability of the proposed model for various kinds of image processing problems, including image enhancement, image restoration, and image segmentation. In addition, the benefits resulting from application of the proposed model have been presented.

The proposed model requires label information of image pixels in advance; however, in some situations this label information may be unavailable or undetermined. Sometimes, it might be necessary to find the actual hidden label properties, and our future work is to refine the proposed model to an unsupervised one, which can automatically analyze and determine the label information for further use. On the other hand, we are trying to tailor this general framework to a particular case. The specialization of the proposed model will involve more issues such as generation of raw image features, transformation of label image properties, integration of different masks, and cooperation of existing methods.

REFERENCES

Kun-Che Lu (呂坤哲) received a B.E. degree and an M.S. degree in Computer Science from Feng Chia University, Taiwan, in 2000 and 2002, respectively. Mr. Lu is currently a Ph.D. student at the Department of Information Engineering and Computer Science, Feng Chia University. His research interests include data mining, image processing, and databases.

Don-Lin Yang (楊東麟) received a B.E. degree in Computer Science from Feng Chia University, Taiwan, in 1973, an M.S. degree in Applied Science from the College of William and Mary in 1979, and a Ph.D. degree in Computer Science from the University of Virginia in 1985. He worked at IBM Santa Teresa Laboratory from 1985 to 1987 and at AT&T Bell Laboratories from 1987 to 1991. Since then, he joined the faculty of Feng Chia University and is currently a professor in the Department of Information Engineering and Computer Science. His research interests include image processing, data mining, and software engineering. Dr. Yang is a member of the IEEE Computer Society and the ACM.